DOCUMENT RESUME

ED 183 198 IR 008 140

AUTHOR	Muiznieks, Viktors J.: Cox, John
TITLE	Documentation of CTRS--Computerized Test-Result Reporting System. The Illinois Series on Educational Application of Computers, No. 22e.
INSTITUTION	Illinois Univ., Urbana. Dept. of Secondary Education.
SPONS AGENCY	EXXON Education Foundation, New York, N.Y.
PUB DATE	79
NOTE	72p.: Computer printout lists may not reproduce: see IR 008 123-139, IR 008 142, and IR 008 144.
EDRS PRICE	MF01 Plus Postage. PC Not Available from EDRS.
DESCRIPTORS	*Computer Managed Instruction: *Computer Programs: Guides: Item Analysis: *Program Descriptions: Programers: Programing Languages: *Recordkeeping: Student Testing: *Test Results
IDENTIFIERS	*BASIC Programing Language

ABSTRACT

The Computerized Test-Result Reporting System (CTRS), which consists of three programs written in the BASIC language, was developed to analyze obective tests, test items, test results, and to provide the teacher-user with interpreted data about the performance of tests, test items, and students. This paper documents the three programs from the viewpoints of both the potential user in a school system who is not necessarily familiar with computers, and the computer programmer wishing to examine the structure of the system or seeking to modify the programs for use on a computer system using a different version of BASIC language. In this paper, the programs are described as they were written in Version 1.0 of the ICOM-DEBBI brand of BASIC. The three programs are PREP, a preparation program used to convert unformulated student answers into a more readily graded form: GRADE, a test grading and analysis program used to produce student results, item analysis, and test reliability results, and to output this information in printed form: and DUMP1, which is used to obtain cumulative statistics about CTRS use for more detailed analysis. Appendices include sample output from GRADE and DUMP1, as well as program listings, variables, and flow diagrams for the three programs, and the TEXT file. (Author/CMV)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

U.S. DEPARTMENT OF HEALTH, EDUCATION & WELFARE
NATIONAL INSTITUTE OF EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

ED183198

Number 22e

The Illinois Series on Educational Application of Computers

DOCUMENTATION of CTRS--
COMPUTERIZED TEST-RESULT REPORTING SYSTEM

Viktors J. Muiznieks and John Cox

"PERMISSION TO REPRODUCE THIS MATERIAL IN MICROFICHE ONLY HAS BEEN GRANTED BY

J Richard Dennis

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

College of Education
Department of Secondary Education
University of Illinois
Urbana, Illinois

This work was supported by a grant from
The EXXON Education Foundation

Copyright, 1979, Department of Secondary Education
University of Illinois

IR008140

To the reader,

This is one of a series of documents prepared as resources for the preservice and inservice training of teachers. The general theme of all items in this series is "teaching with or about computers". The booklets are the collective work of a variety of staff and students (both present and former) of the College of Education, University of Illinois at Urbana-Champaign. The project generously was supported between 1977 and 1979 by a grant from the EXXON Education Foundataion. All responsibility for the content of these documents rests with the authors and does not reflect necessarily the views of the EXXON Foundation. It was my extreme pleasure to direct this effort throughout.

J. Richard Dennis, PhD
Department of Secondary Education
University of Illinois

TABLE OF CONTENTS

Section	Pages
Introduction	1
User's Guide	2
I. Setting Up the Data Files	2
II. Load	3
III. PREP	4
Grading Previously Used Tests	7
Notes on PREP	8
IV. GRADE	8
V. DUMP1	8
Programmer's Guide	10
File Structure	10
Size Considerations	12
Conversion (Transportability) Guidelines	14
Appendices	18
output from GRADE	19
Output from DUMP1	27
Program Listings, Variables, and Flow Diagrams	31
TEXT File	53

INTRODUCTION

The test grading system described in this paper was developed by Viktors J. Muiznieks as part of his PhD dissertation* at the University of Illinois, Urbana-Champaign, in 1978. The programs collectively are known as CTRS (Computerized Test-result Reporting System). CTRS was developed to analyze objective tests, test items, and test results, and to provide the teacher-user with interpreted data about the performance of the tests, test items, and students.

The CTRS package consists of 3 programs all written in the BASIC language. This paper documents these programs from the view of the potential user in a school system who is not necessarily familiar with computers, and from the view of the computer programmer wishing to examine the structure of the system or seeking to modify the programs for use on a computer system using a different version of the BASIC language. This paper describes the programs as they were written in Version 1.0 of the ICOM-DEBBI brand of BASIC. The names and functions of the programs in CTRS are:

PREP--preparation program used to convert "unformatted" student answers into a more readily-graded form

GRADE--test grading and analysis program used to produce student results, item analysis, and test reliability results, and to output this information in printed form.

DUMP1--used to obtain cumulative statistics about CTRS use for more detailed analysis

*Muiznieks, Viktors Janis, "The Effects of Computerized Test-Result Reporting on Teachers' Use of Test Results", Unpublished Doctoral Thesis, Department of Secondary Education, University of Illinois, Urbana, Illinois, July, 1978.

This manual is intended as a guide to the use of the programs of CTRS. We assume that the person running these programs is able to power up his/her computer and to load the computer's operating system. Further background in programming is helpful in understanding how to use CTRS most effectively, but is not necessary to the use of this manual. Of course, the system commands and procedures may be different on other computers, so this manual will serve only as a rough guide which should be changed according to the system that is being used.

I. SETTING UP THE DATA FILES

Before running the programs, it is necessary for the user to obtain a diskette containing the programs and to check on the status of the other files required by the system. These checks should be made prior to attempting to use CTRS:

1. If a file exists on the diskette by the name of RAW, it must be deleted before reading the cards containing student answers. This file is created during the reading process and no other file by that name can exist before the cards are read.

2. Check the existence of each of the following files:
   - SCORE (used to store test answers and grading key)
   - DUM (used to store data temporarily in updating CUM)
   - TEXT (used to house report messages to be printed by the GRADE program)
   - CUM (used to store test statistics on every set of tests graded by CTRS)
   - SDATA (necessary only if the answer key to the test is be saved for future use)

   Some of these files must be sized according to the amount of data being processed (in particular, SCORE and SDATA). A formula which can be used to determine roughly how large files should be is:

   ```
   S*(N + 30)/80 + N*(A + 3)/80
   ```

   where S=number of students taking the test
   N=number of questions on the test
   A=maximum number of possible answers for any question

   The answer is the approximate number of 256 character sectors which should be reserved on the diskette for each file.

II. LOAD--card format and inputting student answers

The input of student answers is assumed to be mark-sensed cards. These cards are similar to IBM keypunch cards, but are designed to permit students to mark answers (numbers, letters, and punctuation) using a soft lead pencil to encode their response (most characters require only 1 or 2 marks in a column to represent). The cards are then read by a card reader able to read mark-sensed cards.

Each student marks his/her name on one card. The cards used generally have 40 columns each of which can contain a character. (See also note 1 under PREP.) The student takes the objective test as usual. However, the answers are marked on the cards instead of the test paper. The computer system on which CTRS was developed imposes two restrictions on the marking of cards:

1. No card, except the end card which the user supplies, should have an & (encoded by a single mark in the top row) marked in column one since that character signals end-of-file and no further data will be read.
2. All preceeding blanks on a card are ignored. Therefore, each card should contain something other than the character & in column one. If column one is left blank, all answers will be shifted by at least one answer (e.g., the real answer to question two is interpreted as the student's answer to question one).

The user should check the order and format of the card deck before trying to load them into the computer. For example, a test with 110 questions would require each student to turn in a name card and three answer cards and the user should check for blanks or the character & in column one of each card. The assumed card order in this case is shown on the next page.

```
First card--> student 1 name card
              student 1 answer card 1 (answers to
              questions 1-40)
              student 1 answer card 2 (answers to
              questions 41-80)
              student 1 answer card 3 (answers to
              questions 81-110)
              student 2 name card
              student 2 answer card 1
              student 2 answer card 2
              student 2 answer card 3
              student 3 name card
              student 3 answer card 1
              student 3 answer card 2
              student 3 answer card 3
              (same order for the rest
              of the students
last card---> (contains & in column one to signal
              end-of-deck to the computer)
```

To read the cards, the deck should first be checked for order and other defective markings. Once the deck is prepared, the cards should be placed in the card reader hopper and read into a file called RAW which is located on the floppy disk drive designated Drive 1.

III. PREP--inputting correct answers and preparing data to be graded

Once the computer has read in the student answers from the cards and has stored them in the data file, RAW, the user is ready to run the PREP program which converts the answers into a format suitable for grading. The user must have an answer key for the test and be prepared to enter the necessary information about the test before attempting to run PREP.

Once the Prep program is running the computer will print the question "ENTERING DATA FOR NEW TEST?" If this is the first time the test has been graded, the user should type "Y" (see section below for grading a second group of answer cards to the same test).

The user then should be prepared to enter the following data as requested:

1. The number of questions on the test.
2. The test-heading data (teacher's name, subject or class name, class period, test date).
3. The maximum number of alternatives for any question on the test. (True-false questions have only 2 alternatives, whereas multiple choice items may have more.)

4. The names used for the alternatives (numbers, letters, etc.). These must correspond to the answering used by the students on their answer cards.
5. The correct answer to each question.
6. The item difficulty estimates (if the teacher has attempted to estimate how difficult each question is).

The computer will prompt the user for each type of information as discussed below.

When the message "HOW MANY QESTIONS ON THE TEST?" is printed, the user should provide the total number of questions. Next, the computer requests "HEADING INFORMATION (TEACHER NAME-CLASS NAME-PERIOD-DATE)" and the user should type the data (which should consist of 72 or fewer characters--see note 2). A sample run shown below indicates how this information is entered for a 50 question test in Mr. Cox's 2nd period Algebra class given on May 2, 1979 (no commas can appear in the heading). The material underlined in this example is typed by the teacher-user.

```
ENTERING DATA FOR NEW TEST (Y OR N) ? Y
HOW MANY QUESTIONS ON THE TEST ? 50
HEADING INFORMATION (TEACHER NAME-CLASS NAME-PERIOD-DATE)
? MR. COX-ALG. I-2ND-05 02 79
```

The computer next asks for the "MAXIMUM NUMBER OF ALTERNATIVES IN ANY ONE QUESTION". The user should type the number of possible answers for the question which has the largest number of alternatives. Finally, the specific names of alternatives for each question are requested. The computer prints "NUMBER OF FIRST QUESTION WHICH HAS DIFFERENT NAMES FOR THE ALTERNATIVES THAN QUESTION NUMBER 1 (0 IF NONE) ?" If all the questions on the test have the same answer alternatives (such as A, B, C, and D) then the user can respond by typing 0 (zero, not oh). If some of the questions vary, the user should enter the number of the first question which has different alternative names (or a different number of alternatives) from those used in question 1. For example, if the test contains 10 true-false questions (student marks "T" or "F"), followed by 8 multiple choice questions (all with 4 alternatives: A, B, C, and D), and the last group of questions each has five possible answers (A, B, C, D, and E), the user would enter the following data:

```
MAXIMUM NUMBER OF ALTERNATIVES IN ANY ONE QUESTION? 5
NUMBER OF FIRST QUESTION WHICH HAS DIFFERENT NAMES FOR THE
ALTERNATIVES THAN QUESTION NUMBER 1 (0 IF NONE) ? 11
NUMBER OF ALTERNATIVES IN QUESTION NUMBER 1 ? 2
WHAT IS ALTERNATIVE NUMBER 1 CALLED? T
WHAT IS ALTERNATIVE NUMBER 2 CALLED? F
NUMBER OF FIRST QUESTION WHICH HAS DIFFERENT NAMES FOR THE
ALTERNATIVES THAN QUESTION NUMBER 11 (0 IF NONE)? 19
NUMBER OF ALTERNATIVES IN QUESTION NUMBER 11? 4
```

```
WHAT IS ALTERNATIVE NUMBER 1 CALLED?  A
WHAT IS ALTERNATIVE NUMBER 2 CALLED?  B
WHAT IS ALTERNATIVE NUMBER 3 CALLED?  C
WHAT IS ALTERNATIVE NUMBER 4 CALLED?  D
NUMBER OF FIRST QUESTION WHICH HAS DIFFERENT NAMES FOR THE
ALTERNATIVES THAN QUESTION NUMBER 19 (0 IF NONE)?  0
(note: all other questions have 5 alternatives)

NUMBER OF ALTERNATIVES IN QUESTION NUMBER 19?  5
WHAT IS ALTERNATIVE NUMBER 1 CALLED?  A
WHAT IS ALTERNATIVE NUMBER 2 CALLED?  B
WHAT IS ALTERNATIVE NUMBER 3 CALLED?  C
WHAT IS ALTERNATIVE NUMBER 4 CALLED?  D
WHAT IS ALTERNATIVE NUMBER 5 CALLED?  E
(see note 3)
```

To minimize the work necessary to enter this information, it is helpful to group questions on the test by number and name of alternatives rather than interspersing different types of questions. For example, place all true-false questions in one section, all questions with alternatives A, B, C, and D in another section, etc. Another step which may be helpful to students in marking their cards is to replace letters (which require 2 marks per column to represent) by numbers (which require only 1 mark per column). This can be done by instructing students to mark 0 for false and 1 for true and numbering the alternatives for multiple choice questions rather than using letters.

Once the user has specified the number and names of alternatives for all questions, the user is given an opportunity to "ADD OR CHANGE ANY ALTERNATIVE NAMES (Y OR N)?" If the user made a mistake, the question number can be entered and the computer will ask which alternative number you want to add or change and will accept the corrections. For example, if the user overlooked the fact that question number 16 had 5 possible names (A, B, C, D, and E), this could be corrected as follows:

```
ADD OR CHANGE ANY ALTERNATIVE NAMES (Y OR N)?  Y
QUESTION NUMBER (TYPE 0 FOR NONE)?  16
ALTERNATIVE NUMBER (TYPE 0 FOR NONE)?  5
NEW NAME FOR ALTERNATIVE  5 IN QUESTION  16?  E
ADD OR CHANGE ANY ALTERNATIVE NAMES (Y OR N)?  N
```

Next the computer requests the user to "ENTER CORRECT ANSWER & DIFFICULTY ESTIMATE FOR EACH QUESTION LEGAL DIFFICULTY ESTIMATES ARE H, M, E, and ?" The user should type in the correct answer, a comma, followed by the estimate (or ? if no estimates are provided). This must be repeated for each question on the test. After these are entered, the computer gives the user an opportunity to correct any typing errors. ("CHANGE ANY ANSWERS OR ESTIMATES (Y OR N)?") If the user made a mistake, the number of

the question can be entered and the computer will accept the corrected information. Part of a sample run is shown below:

```
ENTER CORRECT ANSWER & DIFFICULTY ESTIMATE FOR EACH
QUESTION
LEGAL DIFFICULTY ESTIMATES ARE H, M, E and ?
1? T,M
2? T,M
3? F,E
4? F,M        (user made error--answer should be T)
5? F,E
 .
 .
 .
50? C,E
CHANGE ANY ANSWERS OR ESTIMATES (Y OR N)? Y
WHICH QUESTION DO YOU WANT TO CHANGE (TYPE 0 FOR NONE)? 4
CORRECT ANSWER AND DIFFICULTY ESTIMATE FOR QUESTION 4? T,E
CHANGE ANY ANSWERS OR ESTIMATES (Y OR N)? N
```

The computer will then sort the student answers and store the correct answer data in a file called SCORE (which must be on the diskette in drive 1). This may take a few minutes and the computer will print a message asking you to be patient.

Finally, the computer asks "STORE DATA FOR ANOTHER CLASS (Y OR N)?" The user should respond "Y" if the same test (or a make-up test) will be used at a later time so the information need not be entered again for this second grading. (If the user replies "Y", the computer will need file SDATA on the diskette in the drive designated as drive 1--see note 4.) If the test is not to be graded again, the user can reply "N".

GRADING PREVIOUSLY USED TESTS

Once the answer key has been entered using the PREP program, the user need not enter it again for grading make-up tests or the same test for a different class provided the data is stored as discussed in the previous paragraph. The user merely answers "NO" to the query at the beginning of PREP "ENTERING DATA FOR NEW TEST?" and the program will retrieve the answer key from the file SDATA (see note 5). This can save a great deal of time and effort and, when possible, the user should try to save the answer key (unless it is known that the same test will never be used again).

NOTES ON PREP

1. LIne 1610 should be retyped if fewer than 40 columns are used on a given card. For example, if the teacher had students mark only the first 30 columns of each card, the user should type:

   1610 DATA 30

   before RUNning the program.
2. No commas can be used in the test heading information because of the version of BASIC used. It also requires that no line of input be more than 72 characters in length.
3. PREP is only capable of handling one character answers (such as A, B, T, F, 1, 2, etc.). No provisions are made for grading tests which allow more than one correct answer per question.
4. Before running PREP it is advisable to check file SDATA to see if information is already stored there, since this info will be destroyed when new info is stored. If useful information is already in SDATA, that file should be copied to another diskette.
5. When the user wants to grade another set of tests with a previous answer key, he/she must be sure the correct key is contained in SDATA.

IV. GRADE--grading the answers and obtaining analysis

Once the PREP program is finished, the user is ready to run the GRADE program which will score the tests, print a summary for each student which contains the problems missed and then correct answers, and print an analysis of the class results and test reliability. These results are printed on the system's printer and the paper should be rolled to the top of the next page before starting the program. A sample of the types of statistics which can be obtained from the GRADE program is given in Appendix A. No operator intervention is required during the execution of the program as the results are calculated and printed automatically.

V. DUMP1--obtaining statistics for all tests processed

The DUMP1 program can be run at any time to produce statistics for every test graded by the CTRS. These statistics include the same test analysis and reliability material as produced by GRADE, but the user can specify these results for more than one test and make comparisons of test effectiveness.

Once DUMP1 is running the user will be given four alternatives: 1. EVERYTHING, 2. AVERAGES ONLY, 3. ONE TEACHER ONLY, or 4. STOP. The first option will print out all statistics which have been stored by CTRS so far. The second option prints

out only the average values for these statistics. The third option prints out the statistics for a given teacher--the user will be asked to enter the teacher's name and only that teacher's test statistics will be printed. The fourth option will end the program. Before typing in the number of the option which is desired, the user should again advance the paper in the printer to the top of the next page so the results will be properly paged. A sample of the types of outputs from DUMP1 is found in Appendix B.

PROGRAMMER'S GUIDE

This section of the manual goes into more detail concerning the actual structure of the programs and files used by CTRS and provides some guidelines for adapting them to your own computer system. Program listings are found in Appendix C and the next few sections consider the questions of file structure, transportability or conversion to other versions of BASIC, and the memory size required to use CTRS.

File Structure

CTRS was designed on the IMSAI 8080 microcomputer system running with ICOM'S FDOS-III/DEBBI software. This version of BASIC allows only one input file and one output file to be opened and usable at any time. Further, these files are sequential in nature and records cannot be accessed randomly.

Six data files are needed by CTRS. These files and their functions are given below:

1. RAW--contains "unformatted" student names and answers as input from the card reader
2. SCORE--file containing "formatted" test-grading information including answer key
3. SDATA--contains only the answer key and format information for the test if the user requests to save it
4. TEXT--contains textual information printed by the GRADE program for test evaluation
5. CUM--contains cumulative statistics on all tests graded by CTRS
6. DUM--scratch file used for temporary storage of information

The data from student answer cards is read into the computer by the operator and placed in file RAW. The format of file RAW is as follows:

RECORD NUMBER	DATA
1	Student name
2	First answer card
3	Second answer card (or next student name if only one answer card is used for each student)
.	
.	
.	
	Last answer card for last student
N	END-OF-FILE marker

RAW must be in this format in order for PREP to process the data correctly. After running the CTRS programs for one set of tests, the user must delete RAW before trying to run another set. Other computer systems may require the user to issue other commands: possibly one to delete the file from the directory and one to compact or pack the diskette (removing the gap where RAW previously was stored).

The SCORE file is formatted by the PREP program--enough space must be provided to allow storage of all the data in RAW plus the grading key (correct answers, alternatives, and other info). See the section on size considerations below for more detailed information on how large files should be. The SCORE file's structure is as follows:

RECORD NUMBER	DATA
1	Number of questions on test
2	Heading info (teacher, class, date, etc.)
3	Answer to question 1
4	Difficulty estimate for question 1 (coded as a number)
5	Answer to question 2
.	
.	
.	
	Answer to last question
	Difficulty estimate for last question
	Number of students taking test
.	
.	
.	

The rest of the file consists of possible alternatives for each question (one per record) and the actual student names and answers. If a question does not have the maximum number of alternatives possible, the last alternative is followed by an end of alternatives mark (***). Questions which have letters rather than numbers for alternatives are translated into numbers. For example, if a question has alternatives A, B, C, and D and the student responded with C, the computer would store the number 3 as the answer and the numbers 1, 2, 3, and 4 as the possible alternatives. Those answers which are not chosen from the possible alternatives are translated to 0 and the student's incorrect response is also written in the file as a separate record following the 0.

In this conversion process, the PREP program ignores extra answers on cards, and supplies "blank" answers for any unanswered questions. If a student marks more than one answer to a question, a "random" answer will be supplied. The random answer has a 1 out of 256 chance of being correct.

File SDATA is used to store only the grading key and has the same format as file SCORE except the students' names and answers are not stored. Therefore, it need not be as large as SCORE. The PREP program writes over any information previously placed in SDATA if the user requests that the grading key be stored. It is advisable to copy SDATA to a different diskette if the user wants to save the grading key for another time so that no one accidentally will write over the data while grading a different test.

A copy of the TEXT file is included in Appendix D. It is used by the GRADE program to print out a more meaningful analysis of the test for the teacher.

CUM and DUM are used by GRADE to keep cumulative statistics on tests graded by CTRS. DUM is a scratch file, which must be at least as large as CUM, and in which the updated statistics are stored. Each test's heading is stored, followed by one record for each of 13 data categories: test reliability, standard deviation, standard error of measurement, percentage of positive distractors, percentage of questionable distractors, percentage of inefficient distractors, percentage of incorrect item difficulty estimates, average item difficulty, percentage of exceptionally good questions, percentage of good questions, percentage of marginal questions, percentage of non-discriminating questions, and the percentage of possibly ambiguous questions. Tests are stored in the order in which they are graded and this information is stored for each time the CTRS programs are run--even if only one test is graded.

Size Considerations

The size of the data files used by CTRS will vary depending on the number of tests graded (in the case of CUM) and the number of students and questions on any given test (in the case of SCORE, RAW, and SDATA). To figure how large the SCORE file should be, use the formula

$$(180 + 80N + 10AN + MA)/250$$

where N is the number of test papers being graded, A is the number of questions on the test, and M is the maximum number of alternatives for any question on the test. File SDATA may be slightly smaller since the student data is not stored there. The CUM file should be made as large as possible so it can store information on all the tests which are graded by CTRS. (If the file gets full, the programs will abort and you will have to read the data again.) File DUM should be at least as large as file CUM.

Another important consideration for the programmer interested in adapting these programs to a different computer is the amount of memory which is required for the programs to run. The size and number of tests which the CTRS programs can grade (at one time) is limited by the amount of computer memory available. For the IMSAI 8080 microcomputer on which these programs were developed, 48K of RAM (Random-Access Memory) were available (49,152 8-bit words). The version of BASIC which was used requires 15,735 words of memory and the GRADE program (which is the longest of the CTRS programs) requires 7,785 words for storage. (This is after all comments and spaces are removed and using multiple-programming statements to reduce the amount of line numbers required.)

In addition to the space required for the programming statements, space is also needed to store data values in variables and arrays. In our version of BASIC an integer variable requires 5 words, single precision variables require 7 words, string variables require 6 words, integer arrays require 2 times the number of elements plus 6 plus 2 times the number of dimensions in the array, single precision arrays require 4 times the number of elements plus 6 plus 2 times the number of dimensions, and string arrays require 3 times the number of elements plus 6 plus 2 times the number of dimensions.

In addition to the above constraints, the BASIC used in CTRS requires that storage space be set aside for handling strings (this is not the case on many other computer systems). Although the program can run with 1000 words reserved, 5000 words is better for time considerations because the BASIC interpreter uses a slow routine to retrieve currently unused string space when the (physically) last word allocated to string-handling is used.

For these reasons, an estimate can be obtained of how much memory will be needed to grade a particular set of tests. If we let A represent the number of questions on the test, N the number of people taking the test, and M the maximum number of alternatives on one question, then the short, uncommented version of GRADE will run properly if 11A + 13N + 5AM + 2AN + 5M = 20,178. For example, consider a class of 30 students (N=30) taking a test of 100 questions (A=100) with the longest question having five possible alternatives (M=5). In this case, we can figure that 11*100 + 13*30 + 5*100*5 + 2*100*30 + 5*5 = 10015 which is less than 20178. So the GRADE program would be able to handle this group of tests with plenty of room to spare.

If less memory is available or a different version of BASIC is used, then this formula will not apply, but the programmer should be able to give at least a rough estimate of how much memory is necessary to grade a given test.

1.1

Conversion (Transportability) Guidelines

Unfortunately, there is no one standardized version of the BASIC programming language. Because different versions of BASIC exist, it often is difficult to convert or transport programs from one computer to another. Changes usually need to be made to the syntax of several programming statements before the programs will work. The CTRS programs have been written using rather elementary BASIC statements whenever possible. Even so, there still are several changes that may need to be made depending on the particular version of BASIC which is being used. Typically, these changes need to be made to programming statements involving data files and string handling.

The following list is intended to serve as a guide in identifying statements which might need revision. In the discussion N and N1 are assumed to be numeric values, S$ and S1$ are assumed to be strings or string variables, and LN is assumed to represent a line number within the program.

CLEAR N sets aside N words for handling strings during the course of program execution. Many BASIC'S do not require this statement and it can be left out. Some versions of BASIC require each string to be separately dimensioned which serves the same purpose as this command in reserving string space.

WIDTH N informs BASIC that the output device (printer, terminal, etc.) may contain up to N characters per line. Some BASIC'S use a MARGIN statement or leave the width of the line up to the software to assure that lines do not run over.

OPENR S$:N opens file S$ for reading from the diskette in drive N. Many BASIC'S use statements such as FILE # N,S$ to open a file for reading on channel N. In these BASIC'S no distinction is made between files being used for writing (at the time the files are opened) and those used for reading. The CTRS programs assume only one file to be open for reading and one for writing at any given time (a peculiarity of the version of BASIC used).

OPENW S$:N opens file S$ for writing on the diskette in drive N. (See the discussion above.)

CLOSR closes the file currently being read. (See the discussion below).

CLOSW closes the file currently being written. Some BASIC'S have no statement equivalent to CLOSR or CLOSW. Other versions use a RESTORE # N statement.

DSKIN N (or S$) reads the next numeric (or string) record from the file open for reading. Some BASIC'S use statements such as READ # N,N1 or INPUT # N,S$. Note that when reading a record into a string variable (using either the INPUT or DSKIN statements) the CTRS programs first set the string variable equal to the null string (""). This is done to ensure that a null string will be correctly read. In our version of BASIC, if a null record is read, the value of the string variable is not altered from its previous value. In most BASIC'S there will be no need to first set the value of the string variable to the null string.

DSKOUT N (or S$) writes numeric (or string) records into the file open for writing. Some BASIC'S use statements like PRINT # N,N1 or WRITE # N,S$.

ON ERROR GOTO LN is used to test for end-of-file condition for reading from diskette. If an error (end-of-file) occurs during an attempted read, control is transferred to the statement numbered LN. Some BASIC'S use statements such as IF EOF THEN LN. This problem can be overcome by having the software keep track of how many data items are in the file and how many have been read so that the program never requests a read unless you are sure data still exists in the file.

IF ERR=27 THEN LN is used in conjunction with the ON ERROR statement to verify that the error which occurred really was an end-of-file. In our version of BASIC, the system variable ERR is set to a numeric value of 27 if an end-of-file error occurs. If no ON ERROR statement is available in your version of BASIC, this statement can be replaced with GOTO LN.

RESUME LN is used after an ON ERROR statement to resume program execution at the statement numbered LN. If no ON ERROR statement is used, this statement can be replaced with GOTO LN (or left out entirely providing you move the rest of the program up one line number).

STR$(N) is a function which returns the string representation of the number N. If STR$ or its equivalent is not available, replace STR$(N) with N. In the version of BASIC on which the CTRS programs were developed, the PRINT statement automatically places a blank before and after each number printed. The STR$ function is used to control these blanks. Note that a string value of blank (" ") may have to be inserted in PRINT statements before and after numeric variables to achieve the same spacing as that appearing in the CTRS program printouts.

MID$(S$,N,N1) is a function which returns the substring of string variable S$ beginning at position N, having length N1. If N1 is omitted, it returns the rest of the string starting at position N. This function often is called SUBSTR in other BASIC'S

or is denoted differently (for example, S$(N,N1) is used in some versions of BASIC which do not allow string arrays.)

FNA(N,N1)= denotes the definition of a user-defined function. If single-line user-defined functions are not allowed, replace each occurrence of all function calls with their equivalents by substituting the actual function in the assignment statement which contains the call.

DEFINT is used in our version of BASIC to define variable types by the first letter. This statement is used in the GRADE program to define most of the variables as integer types. If memory size poses no problem or if your system does not permit this, the statement can be left out. Some BASICS have a predetermined method of defining variable types.

ERASE eliminates arrays which are no longer needed or which have to be redimensioned before using again. This command saves space, but some versions of BASIC do not offer this feature.

RESTORE allows DATA statements to be reread starting from the beginning. It is necessary for a second run of the program to start from the top. If your version of BASIC does not have this feature, add another set of duplicate DATA statements.

INSTR(S$,S1$) searches for the first occurrence of string S1$ within S$ and returns the starting position. If this function is not available, a loop can be used to replace it.

LPRINT is the same as a regular PRINT, but sends the data to the computer's printer port. If none is available or your version of BASIC does not have this command, a paper printing terminal can be used by replacing these commands by PRINT.

TAB(N) causes the printing device to skip to the Nth column and resume printing there. This command can be replaced with a string of spaces if column format is not important. If column format is necessary, a loop can be inserted to print blank spaces until the Nth column is reached.

SQR(N) returns the square root of N. Some BASIC'S use SQRT or other names for the same function.

Note also that the BASIC used must permit string arrays and execution time dimensioning of arrays. If execution time dimensioning of arrays is not possible, set array sizes to the maximum size possible. If string arrays are not permitted, the programmer will have to devise some method of making long arrays usable in smaller form (perhaps by using the substring function provided with the software).

The file preparation program, PREP, and the DUMP1 program make use of IF-THEN statements having programming statements (not just line numbers) following the THEN, and also logical OR comparisons in IF-THEN statements. If the BASIC being used does not support such features, these statements can be rewritten by breaking the original IF-THEN statements into two or more equivalent IF-THEN statements which direct the execution to the proper conclusion.

AN ADDITIONAL NOTE

In the original version of CTRS a BATCH program was included to help the user in executing PREP and GRADE without entering all the necessary system commands. Since it is machine specific, it is not included in this manual. Most systems either allow the programmer to construct procedure files to accomplish this or allow chaining from one BASIC program to another. If this feature is desirable, the programmer should write such a program or include chaining in the version of CTRS which is written for the specific system. Many teachers or users may not be too acquainted with operating a computer and any steps which can be taken to make the CTRS package easier to operate will be appreciated and greatly encourage its use.

APPENDIX A

SAMPLE OUTPUT FROM GRADE

STUDENT PRINTOUTS: EACH STUDENT RECEIVES A SUMMARY OF QUESTIONS MISSED INCLUDING CORRECT ANSWER ALONG WITH HIS/HER INCORRECT RESPONSE, RAW SCORE, AND PERCENTAGE CORRECT.

```
MR. COX-ALG. I-2ND-05 2 79
HATCHET HI

NO QUESTIONS MISSED
SCORE =  50  OUT OF  50
PERCENTAGE CORRECT =  100


MR. COX-ALG. I-2ND-05 2 79
MANCHU FU
YOU MISSED THE FOLLOWING QUESTION(S).  YOUR ANSWER IS PRINTED FIRST, FOLLOWED BY THE CORRECT ANSWER.
 11. (4)-->(3)
SCORE =  49  OUT OF  50
PERCENTAGE CORRECT =  98


MR. COX-ALG. I-2ND-05 2 79
HIGH SALLY
YOU MISSED THE FOLLOWING QUESTION(S).  YOUR ANSWER IS PRINTED FIRST, FOLLOWED BY THE CORRECT ANSWER.
 11. (4)-->(3)     28. (2)-->(1)     33. (6)-->(5)     40. ( )-->(3)     50. (2)-->(3)
SCORE =  45  OUT OF  50
PERCENTAGE CORRECT =  90
```

19

SCORE DISTRIBUTION: ONE COPY IS PROVIDED FOR THE TEACHER'S USE OR FOR POSTING IN THE CLASSROOM.

MR COX-ALG I-2ND-05 2 79

SCORE	NUMBER OF STUDENTS WITH THAT SCORE (EACH * REPRESENTS A STUDENT)
0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	*
24	
25	*
26	
27	
28	*
29	*
30	*
31	*
32	
33	
34	
35	*
36	***
37	*
38	
39	
40	*
41	
42	**
43	
44	*
45	*
46	
47	
48	
49	***
50	*

CLASS SUMMARY: EACH STUDENT IS LISTED WITH THE PROBLEMS MISSED
AND A WRITTEN SUMMARY OF TEST STATISTICS IS INCLUDED FOR THE TEACHER.

MR. COX-ALG. I-2ND-05 2 79

STUDENT NAME	SCORE	PERCENTAGE	RANK	GRADE	QUESTION(S) MISSED
BOTWINSKI CHRIS	36	72 %	11	( )	1, 2, 6, 11, 12, 13, 28, 30, 31, 34, 35, 36, 39, 50
CATFISH SAM	42	84 %	7	( )	1, 12, 13, 28, 34, 35, 36, 44
COOL JOE	30	60 %	16	( )	1, 5, 7, 12, 13, 15, 18, 20, 22, 23, 25, 27, 28, 30, 32, 35, 39, 43, 44, 48
CREAMCHEEZ SUZIE	44	88 %	6	( )	1, 9, 10, 11, 13, 28
DUCK DONALD	49	98 %	2	( )	28
FERN ART	42	84 %	7	( )	11, 12, 13, 22, 25, 28, 31, 32
FRAN ANDY	40	80 %	9	( )	1, 10, 11, 13, 23, 28, 29, 32, 33, 34
FUNGU PHIL	49	98 %	2	( )	2
HATCHES JOB	25	50 %	19	( )	1, 2, 3, 4, 10, 11, 12, 13, 14, 15, 16, 17, 18, 22, 24, 27, 28, 29, 30, 35 36, 40, 41, 44, 47
HATCHET HI	50	100 %	1	( )	
HATCHET MOLLY	28	56 %	18	( )	1, 3, 9, 10, 12, 13, 14, 15, 19, 22, 24, 28, 29, 34, 39, 42, 43, 44, 45, 46 47, 48
HIGH SALLY	45	90 %	5	( )	11, 28, 33, 40, 50
KUGEL KRIS	36	72 %	11	( )	1, 4, 11, 12, 19, 22, 23, 28, 29, 32, 33, 34, 42, 47
LOW JOE	23	46 %	20	( )	1, 5, 6, 11, 12, 13, 14, 15, 17, 18, 19, 22, 23, 24, 28, 31, 32, 33, 35, 36 39, 41, 43, 47, 48, 49, 50
MANCHU FU	49	98 %	2	( )	11
PUKE NUGEL	29	58 %	17	( )	1, 2, 4, 8, 11, 13, 14, 15, 18, 23, 24, 27, 28, 31, 32, 36, 42, 44, 47, 48 49
SMITH BART	31	62 %	15	( )	12, 13, 14, 15, 27, 28, 29, 31, 34, 36, 37, 38, 39, 43, 44, 45, 46, 47, 48
TULL JETHRO	36	72 %	11	( )	4, 5, 6, 10, 11, 13, 14, 15, 20, 27, 28, 32, 39, 44
TULL TONY	37	74 %	10	( )	2, 9, 11, 13, 15, 23, 25, 27, 28, 29, 32, 34, 36
ZIFFEL ARNOLC	35	70 %	14	( )	1, 10, 13, 15, 18, 25, 28, 29, 34, 36, 40, 42, 44, 45, 46

THE TEST SCORES RANGED FROM 23 TO 50 --- A SPREAD OF 27 POINTS.

THE AVERAGE SCORE IS 38 POINTS.

THE THEORETICAL VALUE OF YOUR TEST RELIABILITY = .95 (OUT OF A POSSIBLE 1.00). (THE AVERAGE RELIABILITY OF TEACHER-MADE TESTS IS ABOUT .60). THEREFORE, YOU PROBABLY CAN ASSUME THAT THE RESULTS OF YOUR TEST ARE VERY RELIABLE.
BEWARE!!! A GOOD TEST SHOULD BE RELIABLE. HOWEVER, JUST BECAUSE A TEST IS RELIABLE DOESN'T MAKE IT GOOD!
A VERY RELIABLE TEST COULD BE TESTING SOMETHING DIFFERENT FROM WHAT YOU WANT TO TEST.

THE SCORES ON THE TEST HAVE A STANDARD DEVIATION OF 8.18 (THIS MEANS THAT ABOUT TWO-THIRDS OF THE SCORES FALL BETWEEN 29 AND 46). FROM THE STANDARD DEVIATION, AND THE TEST RELIABILITY, IT IS THEORETICALLY POSSIBLE TO CALCULATE HOW MANY POINTS EACH STUDENT MAY HAVE MISSED (OR GOTTEN RIGHT) BY CHANCE FACTORS ALONE (SUCH AS GUESSING). FOR THIS TEST, YOU CAN ASSUME THAT EACH STUDENT'S SCORE IS CORRECT TO WITHIN 2 POINT(S).

CLASS SUMMARY CONTINUED

KEY TO STUDENT ANSWER DISTRIBUTION SYMBOLS:

A * INDICATES THE CORRECT ANSWER TO EACH QUESTION

A + INDICATES THAT AN INCORRECT ANSWER WAS AS EXCEPTIONALLY GOOD ALTERNATIVE TO INCLUDE IN THE QUESTION. AT LEAST TWO-THIRDS OF THE STUDENTS WHO CHOSE THE ALTERNATIVE RANKED IN THE BOTTOM-HALF OF THE CLASS.
YOU HAVE 0 ALTERNATIVE(S) MARKED WITH A +. THIS IS 0 % OF ALL THE POSSIBLE INCORRECT ALTERNATIVES ON THE TEST.

A ? INDICATES THAT AN INCORRECT ANSWER (OR THE QUESTION) MAY BE AMBIGUOUS. STUDENTS WHO RANKED IN THE TOP-HALF OF THE CLASS CHOSE THE ALTERNATIVE MORE OFTEN THAN STUDENTS WHO RANKED IN THE BOTTOM-HALF OF THE CLASS.
YOU HAVE 6 ALTERNATIVE(S) MARKED WITH A ?. THIS IS 4 % OF ALL THE POSSIBLE INCORRECT ALTERNATIVES ON THE TEST.

AN 'OMIT' MEANS THAT A STUDENT DID NOT ANSWER THE QUESTION, OR A STUDENT MISMARKED THE ANSWER CARD SO THAT THE ANSWER WAS NOT ONE OF THE POSSIBLE CHOICES.

ITEM ANALYSIS: EACH QUESTION IS ANALYZED FOR DIFFICULTY, DISCRIMINATION, AND RESPONSE FREQUENCY.

MR. COX-ALG. I-2ND-05 2 79

QUESTION	STUDENT RESPONSE DISTRIBUTION	% CORRECT ANS	DIFFICULTY	ESTIMATE OF QUESTION DISCRIMINATION ABILITY
1	0= 11 *1= 9	45 %	HARD (MEDIUM)	GOOD DISCRIMINATION
2	0= 5 *1= 15	75 %	MEDIUM	MARGINAL DISCRIMINATION
3	*0= 18 1= 2	90 %	EASY	EXCEPTIONALLY GOOD DISCRIMINATION
4	0= 4 *1= 16	80 %	MEDIUM (EASY)	EXCEPTIONALLY GOOD DISCRIMINATION
5	*0= 17 1= 3	85 %	EASY	EXCEPTIONALLY GOOD DISCRIMINATION
6	*0= 17 1= 3	85 %	MEDIUM (EASY)	EXCEPTIONALLY GOOD DISCRIMINATION
7	*0= 19 1= 1	95 %	EASY	EXCEPTIONALLY GOOD DISCRIMINATION
8	0= 1 *1= 19	95 %	HARD (EASY)	EXCEPTIONALLY GOOD DISCRIMINATION
9	*0= 17 1= 3?	85 %	MEDIUM (EASY)	QUESTION DISCRIMINATES THE 'WRONG' WAY
10	*0= 14 1= 6	70 %	MEDIUM	MARGINAL DISCRIMINATION
11	1= 4 2= 0 *3= 8 4= 8	40 %	MEDIUM	QUESTION DOES NOT DISCRIMINATE AMONG STUDENTS
12	1= 7 2= 1 *3= 11 4= 1	55 %	MEDIUM	GOOD DISCRIMINATION
13	1= 3 2= 3 3= 8 *4= 6	30 %	HARD (MEDIUM)	GOOD DISCRIMINATION
14	*1= 14 2= 2 3= 2 4= 2	70 %	EASY (MEDIUM)	EXCEPTIONALLY GOOD DISCRIMINATION
15	1= 0 *2= 11 3= 2 4= 7	55 %	MEDIUM	GOOD DISCRIMINATION
16	1= 0 2= 0 *3= 19 4= 1	95 %	EASY	EXCEPTIONALLY GOOD DISCRIMINATION
17	1= 2 *2= 18 3= 0 4= 0	90 %	EASY	EXCEPTIONALLY GOOD DISCRIMINATION
18	*1= 15 2= 2 3= 1 4= 2	75 %	MEDIUM	EXCEPTIONALLY GOOD DISCRIMINATION
19	*1= 17 2= 0 3= 0 4= 0 5= 3	85 %	EASY	EXCEPTIONALLY GOOD DISCRIMINATION
20	1= 2 *2= 18 3= 0 4= 0 5= 0	90 %	EASY	EXCEPTIONALLY GOOD DISCRIMINATION
21	1= 0 2= 0 *3= 20 4= 0 5= 0	100 %	MEDIUM (EASY)	QUESTION DOES NOT DISCRIMINATE AMONG STUDENTS
22	1= 0 2= 0 3= 0 *4= 14 5= 5			
	OMIT = 1	70 %	MEDIUM	GOOD DISCRIMINATION

ITEM ANALYSIS CONTINUED

26	1= 0	2= 0	3= 0	4= 0	*5= 20	100 %	EASY		QUESTION DOES NOT DISCRIMINATE AMONG STUDENTS
27	1= 0	2= 2	3= 1	*4= 14	5= 3	70 %	MEDIUM		GOOD DISCRIMINATION
28	*1= 3	2= 12?	3= 2	4= 1	5= 2	15 %	HARD		EXCEPTIONALLY GOOD DISCRIMINATION
29	1= 1	2= 2	3= 2?	4= 2	*5= 13	65 %	HARD	(MEDIUM)	MARGINAL DISCRIMINATION
30	1= 2	*2= 17	3= 1	4= 0	5= 0	85 %	EASY		EXCEPTIONALLY GOOD DISCRIMINATION
31	1= 2	*2= 15	3= 2	4= 1	5= 0	75 %	MEDIUM		GOOD DISCRIMINATION
32	1= 2	2= 2	*3= 12	4= 3	5= 1	60 %	HARD	(MEDIUM)	MARGINAL DISCRIMINATION
33	1= 0	2= 1	3= 0	4= 2	*5= 16				
	OMIT = 1					80 %	MEDIUM	(EASY)	QUESTION DOES NOT DISCRIMINATE AMONG STUDENTS
34	*1= 12	2= 3?	3= 1	4= 4	5= 0	60 %	HARD	(MEDIUM)	MARGINAL DISCRIMINATION
35	1= 2	2= 0	3= 0	*4= 15	5= 3	75 %	MEDIUM		GOOD DISCRIMINATION
36	1= 2	2= 1	3= 0	*4= 12	5= 5	60 %	EASY	(MEDIUM)	GOOD DISCRIMINATION
37	1= 1	2= 0	3= 0	4= 0	*5= 19	95 %	HARD	(EASY)	EXCEPTIONALLY GOOD DISCRIMINATION
38	*1= 19	2= 1	3= 0	4= 0	5= 0	95 %	HARD	(EASY)	EXCEPTIONALLY GOOD DISCRIMINATION
39	1= 1	*2= 14	3= 2	4= 2	5= 1	70 %	EASY	(MEDIUM)	EXCEPTIONALLY GOOD DISCRIMINATION
40	1= 0	2= 1	*3= 17	4= 1	5= 0				
	OMIT = 1					85 %	EASY		MARGINAL DISCRIMINATION
41	1= 0	2= 0	*3= 18	4= 0	5= 2	90 %	MEDIUM	(EASY)	EXCEPTIONALLY GOOD DISCRIMINATION
42	*1= 16	2= 3	3= 1	4= 0	5= 0	80 %	MEDIUM	(EASY)	EXCEPTIONALLY GOOD DISCRIMINATION
43	*1= 16	2= 2	3= 0	4= 0	5= 2	80 %	MEDIUM	(EASY)	EXCEPTIONALLY GOOD DISCRIMINATION
44	1= 2	2= 1	3= 2	4= 3	*5= 12	60 %	MEDIUM		GOOD DISCRIMINATION
45	1= 0	2= 2	*3= 17	4= 0	5= 1	85 %	EASY		EXCEPTIONALLY GOOD DISCRIMINATION
46	1= 1	*2= 17	3= 2	4= 0	5= 0	85 %	EASY		EXCEPTIONALLY GOOD DISCRIMINATION
47	*1= 14	2= 4	3= 1	4= 1	5= 0	70 %	MEDIUM		EXCEPTIONALLY GOOD DISCRIMINATION
48	1= 0	2= 0	3= 1	*4= 15	5= 4	75 %	MEDIUM		EXCEPTIONALLY GOOD DISCRIMINATION
49	1= 1	2= 1	3= 0	4= 0	*5= 18	90 %	EASY		EXCEPTIONALLY GOOD DISCRIMINATION
50	1= 0	2= 3	*3= 17	4= 0	5= 0	95 %	EASY		MARGINAL DISCRIMINATION

WRITTEN SUMMARY: TO ASSIST THE TEACHER IN ANALYZING THE TEST, THE COMPUTER PROVIDES A WRITTEN SUMMARY OF THE TEST RESULTS.

MR. COX-ALG. I-2ND-05 2 79

OBSERVATIONS ABOUT YOUR TEST

IF NO ONE CHOOSES AN INCORRECT ALTERNATIVE, THEN THAT ALTERNATIVE IS NOT VERY EFFECTIVE. YOU HAD 65 INCORRECT ALTERNATIVE(S) WHICH NO ONE CHOSE. THIS IS 40 % OF ALL THE POSSIBLE INCORRECT ALTERNATIVES ON THE TEST.
49 OF THESE WERE IN EASY QUESTIONS 16 OF THESE WERE IN MEDIUM QUESTIONS 0 OF THESE WERE IN HARD QUESTIONS
YOU SHOULD EXPECT TO HAVE SOME INCORRECT ALTERNATIVES WHICH NO ONE TAKES IN EASY QUESTIONS. HOWEVER, YOU PROBABLY DON'T WANT A LARGE NUMBER OF INCORRECT ALTERNATIVES WHICH NO ONE TAKES IN MEDIUM OR HARD QUESTIONS.

THE DIFFICULTY COLUMN CONTAINS YOUR ESTIMATE OF HOW HARD YOU THOUGHT EACH QUESTION WOULD BE. IF YOU DID NOT ESTIMATE THE DIFFICULTY OF A QUESTION, OR IF IT TURNS OUT THAT THE CALCULATED DIFFICULTY IS NOT THE SAME AS YOUR ESTIMATE, THEN THE CALCULATED DIFFICULTY IS PRINTED IN PARENTHESES.

WRITTEN SUMMARY CONTINUED

AN EASY QUESTION IS ONE WHICH MORE THAN 75% OF THE STUDENTS GET CORRECT.
A MEDIUM QUESTION IS ONE WHICH 25-75% OF THE STUDENTS GET CORRECT.
A HARD QUESTION IS ONE WHICH LESS THAN 25% OF THE STUDENTS GET CORRECT.

YOU ESTIMATED THE DIFFICULTY CORRECTLY ON 58 % OF THE QUESTIONS ON THE TEST.

27 OF THE QUESTIONS ON YOUR TEST WERE EASY. THIS IS 54 % OF ALL THE QUESTIONS ON THE TEST.
22 OF THE QUESTIONS ON YOUR TEST WERE MEDIUM. THIS IS 44 % OF ALL THE QUESTIONS ON THE TEST.
1 OF THE QUESTIONS ON THE TEST WERE HARD. THIS IS 2 % OF ALL THE QUESTIONS ON THE TEST.

THE DISCRIMINATION OF EACH QUESTION IS BASED ON HOW WELL THE TOP-HALF OF THE STUDENTS DO ON THE QUESTION RELATIVE TO THE BOTTOM-HALF OF THE STUDENTS. THE DISCRIMINATION ESTIMATE IS NOT AN ESTIMATE OF HOW WELL THE QUESTION MEASURES KNOWLEDGE, BUT RATHER OF HOW WELL THE QUESTION SEPARATES THE BETTER-SCORING STUDENTS FROM THE POORER-SCORING STUDENTS. THE ESTIMATE TAKES INTO ACCOUNT EACH QUESTION'S DIFFICULTY. HOWEVER, THE ESTIMATE IS BASED ON THE QUESTION AS A WHOLE, AND IT DOES NOT TAKE INTO ACCOUNT THE EXISTENCE OF ANY QUESTIONABLE INCORRECT ALTERNATIVES (INDICATED BY ?'S). THEREFORE, IT IS POSSIBLE TO GET AN 'EXCEPTIONALLY GOOD DISCRIMINATION' ESTIMATE AND STILL HAVE SOME INCORRECT ALTERNATIVES MARKED WITH ?'S.

'EXCEPTIONALLY GOOD DISCRIMINATION' MEANS THAT MOST OF THE STUDENTS WHO GOT THE QUESTION RIGHT WERE IN THE TOP-HALF OF THE CLASS.
YOU HAD 26 QUESTION(S) MARKED AS HAVING EXCEPTIONALLY GOOD DISCRIMINATION. THIS IS 52 % OF ALL THE QUESTIONS ON THE TEST.

'GOOD DISCRIMINATION' MEANS THAT THE QUESTION DOES AN ADEQUATE JOB OF SEPARATING THE BETTER STUDENTS FROM THE POORER STUDENTS.
YOU HAD 10 QUESTION(S) MARKED AS HAVING GOOD DISCRIMINATION. THIS IS 20 % OF ALL THE QUESTIONS ON THE TEST.

'MARGINAL DISCRIMINATION' MEANS THAT THE QUESTION JUST BARELY SEPARATES THE BETTER STUDENTS FROM THE POORER STUDENTS.
YOU HAD 8 QUESTION(S) MARKED AS HAVING MARGINAL DISCRIMINATION. THIS IS 16 % OF ALL THE QUESTIONS ON THE TEST.

'DOES NOT DISCRIMINATE BETWEEN STUDENTS' MEANS THAT ALL STUDENTS DO EQUALLY WELL ON THAT QUESTION (REGARDLESS OF THEIR SCORE).
YOU HAD 5 QUESTION(S) MARKED AS NOT DISCRIMINATING BETWEEN STUDENTS. THIS IS 10 % OF ALL THE QUESTIONS ON THE TEST.

'DISCRIMINATES IN THE WRONG DIRECTION' MEANS THAT THE POORER STUDENTS GOT THE QUESTION RIGHT MORE OFTEN THAN THE BETTER STUDENTS.
YOU HAVE 1 QUESTION(S) MARKED AS DISCRIMINATING IN THE WRONG DIRECTION. THIS IS 2 % OF ALL THE QUESTIONS ON THE TEST.

WRITTEN SUMMARY CONTINUED

IF A QUESTION DOES NOT DISCRIMINATE VERY WELL IT MAY STILL BE A VERY GOOD QUESTION. IT IS POSSIBLE THAT THE QUESTION MEASURES SOMETHING DIFFERENT FROM WHAT THE MAJORITY OF THE TEST MEASURES. HOWEVER, IT MIGHT BE A GOOD IDEA TO LOOK AT SUCH QUESTIONS TO SEE IF THERE MIGHT BE SOME PROBLEMS WITH THEM. ANY QUESTIONS THAT DISCRIMINATE IN THE WRONG DIRECTION SHOULD BE EXAMINED TO SEE IF THEY ARE AMBIGUOUS (OR IF THE ANSWER IS WRONG).

*** WARNING *** BLIND ACCEPTANCE (OR THOUGHTLESS DISREGARD) OF THE ABOVE RESULTS COULD BE HAZARDOUS TO YOUR TESTING PRACTICES!!

THE COMMENTS MADE ON THIS PRINTOUT ARE BASED ON (GENERALLY SOUND) TEST-ANALYSIS THEORIES. THE PURPOSE OF TEST-ANALYSIS IS TO POINT-OUT TO YOU THE POSSIBLE STRENGTHS AND WEAKNESSES OF YOUR TEST. ALTHOUGH IT IS POSSIBLE TO MAKE GENERAL STATEMENTS ABOUT YOUR TEST, YOU ARE THE ONLY ONE WHO KNOWS YOUR STUDENTS, YOUR CLASSROOM, AND THE PARTICULAR CIRCUMSTANCES UNDER WHICH THE TEST WAS GIVEN. THEREFORE, YOU MUST BE THE ONE TO DECIDE WHETHER OR NOT THE COMMENTS MADE ON THIS PRINTOUT ARE RELEVANT TO YOUR PARTICULAR TEST.

ALWAYS KEEP IN MIND THAT THE ABOVE RESULTS ARE NOT ABSOLUTE. TESTS, AND TEST QUESTIONS, DO NOT HAVE UNALTERABLE CHARACTERISTICS. THE SAME TEST, GIVEN TO A DIFFERENT GROUP OF STUDENTS, OR EVEN TO THE SAME GROUP OF STUDENTS ON ANOTHER DAY, COULD HAVE QUITE DIFFERENT RESULTS. TEST-ANALYSIS IS NOT DESIGNED TO HELP YOU BUILD 'PERFECT' TESTS, BUT RATHER TO AID YOU IN GAINING AN UNDERSTANDING OF THE TESTS YOU GIVE. HOPEFULLY, YOU CAN USE THAT UNDERSTANDING TO IMPROVE YOUR FUTURE TESTS.

APPENDIX D

TEXT FILE LISTING

OPTION ONE: A SUMMARY OF TEST STATISTICS IS PRINTED FOR EACH TEST GRADED AND ALL TESTS ARE AVERAGED SO TEACHERS CAN COMPARE THEIR TEST RESULTS WITH THOSE OF OTHERS.

```
MR. COX-ALG. I-2ND-05 2 79
TEST RELIABILITY = .95
STANDARD DEVIATION = 2.35
STANDARD ERROR OF MEASUREMENT = .53
% + ALTERNATIVES = 3
% ? ALTERNATIVES = 0
% 0 ALTERNATIVES = 30
% INCORRECT ITEM DIFFICULTY ESTIMATES = 70
AVERAGE ITEM DIFFICULTY = .4
% EXCEPTIONALLY GOOD QUESTIONS = 50
% GOOD QUESTIONS = 0
% MARGINAL QUESTIONS = 0
% NON-DISCRIMINATING QUESTIONS = 40
% POSSIBLY AMBIGUOUS QUESTIONS = 10

MR. JONES-CHEM. -3RD-05 02 79
TEST RELIABILITY = .98
STANDARD DEVIATION = 2.32
STANDARD ERROR OF MEASUREMENT = .33
% + ALTERNATIVES = 0
% ? ALTERNATIVES = 0
% 0 ALTERNATIVES = 48
% INCORRECT ITEM DIFFICULTY ESTIMATES = 30
AVERAGE ITEM DIFFICULTY = .8
% EXCEPTIONALLY GOOD QUESTIONS = 80
% GOOD QUESTIONS = 10
% MARGINAL QUESTIONS = 10
% NON-DISCRIMINATING QUESTIONS = 0
% POSSIBLY AMBIGUOUS QUESTIONS = 0

MR. COX-ALG. I-2ND-05 2 79
TEST RELIABILITY = .95
STANDARD DEVIATION = 8.18
STANDARD ERROR OF MEASUREMENT = 1.83
% + ALTERNATIVES = 0
% ? ALTERNATIVES = 4
% 0 ALTERNATIVES = 40
% INCORRECT ITEM DIFFICULTY ESTIMATES = 42
AVERAGE ITEM DIFFICULTY = .76
% EXCEPTIONALLY GOOD QUESTIONS = 52
% GOOD QUESTIONS = 20
% MARGINAL QUESTIONS = 16
% NON-DISCRIMINATING QUESTIONS = 10
% POSSIBLY AMBIGUOUS QUESTIONS = 2
```

OPTION ONE OR TWO: THE AVERAGE FOR ALL TESTS GRADED IS PRINTED
WHEN EITHER OPTION ONE OR TWO IS CHOSEN.

AVERAGE VALUES	LOW VALUE	HIGH VALUE	STAN. DEV.
TEST RELIABILITY = .95	.95	.98	.01
STANDARD DEVIATION = 5.25	2.32	8.18	2.92
STANDARD ERROR OF MEASUREMENT = 1.13	.33	1.83	.7
% + ALTERNATIVES = .75	0	3	1.29
% ? ALTERNATIVES = 2	0	4	2
% 0 ALTERNATIVES = 39.5	30	48	6.38
% INCORRECT ITEM DIFFICULTY ESTIMATES = 46	30	70	14.69
AVERAGE ITEM DIFFICULTY = .68	.4	.8	.16
% EXCEPTIONALLY GOOD QUESTIONS = 58.5	50	80	12.43
% GOOD QUESTIONS = 12.5	0	20	8.29
% MARGINAL QUESTIONS = 10.5	0	16	6.53
% NON-DISCRIMINATING QUESTIONS = 15	0	40	15
% POSSIBLY AMBIGUOUS QUESTIONS = 3.5	0	10	3.84

4 TESTS HAVE BEEN GRADED

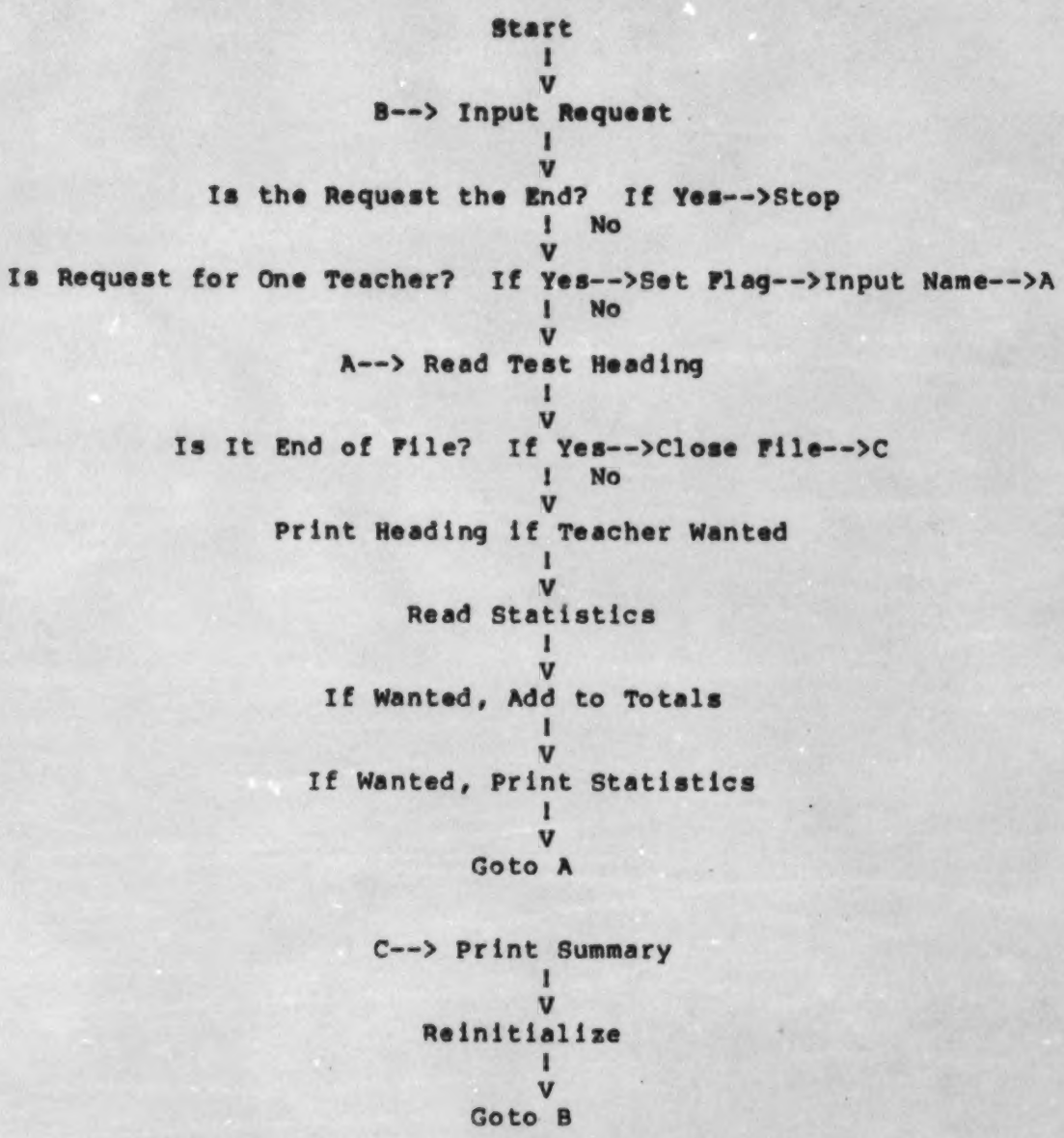
OPTION THREE: THE TESTS FOR ONLY ONE TEACHER ARE DISPLAYED ALONG WITH THE AVERAGE VALUES FOR THOSE TESTS.

```
MR  COX-ALG  I-2ND-05 2 79
TEST RELIABILITY = .95
STANDARD DEVIATION = 2.35
STANDARD ERROR OF MEASUREMENT = .53
% + ALTERNATIVES = 3
% ? ALTERNATIVES = 0
% 0 ALTERNATIVES = 30
% INCORRECT ITEM DIFFICULTY ESTIMATES = 70
AVERAGE ITEM DIFFICULTY = .4
% EXCEPTIONALLY GOOD QUESTIONS = 50
% GOOD QUESTIONS = 0
% MARGINAL QUESTIONS = 0
% NON-DISCRIMINATING QUESTIONS = 40
% POSSIBLY AMBIGUOUS QUESTIONS = 10
```

AVERAGE VALUES	LOW VALUE	HIGH VALUE	STAN. DEV.
TEST RELIABILITY = .95	.95	.95	0
STANDARD DEVIATION = 6.23	2.35	8.18	2.74
STANDARD ERROR OF MEASUREMENT = 1.39	.53	1.83	.61
% + ALTERNATIVES = 1	0	3	1.41
% ? ALTERNATIVES = 2.66	0	4	1.88
% 0 ALTERNATIVES = 36.66	30	40	4.71
% INCORRECT ITEM DIFFICULTY ESTIMATES = 51.33	42	70	13.19
AVERAGE ITEM DIFFICULTY = .64	.4	.76	.16
% EXCEPTIONALLY GOOD QUESTIONS = 51.33	50	52	.94
% GOOD QUESTIONS = 13.33	0	20	9.42
% MARGINAL QUESTIONS = 10.66	0	16	7.54
% NON-DISCRIMINATING QUESTIONS = 20	10	40	14.14
% POSSIBLY AMBIGUOUS QUESTIONS = 4.66	2	10	3.77

```
 3 TESTS HAVE BEEN GRADED
```

The following flowchart is intended to be a guide to the logic used in the DUMP1 program.

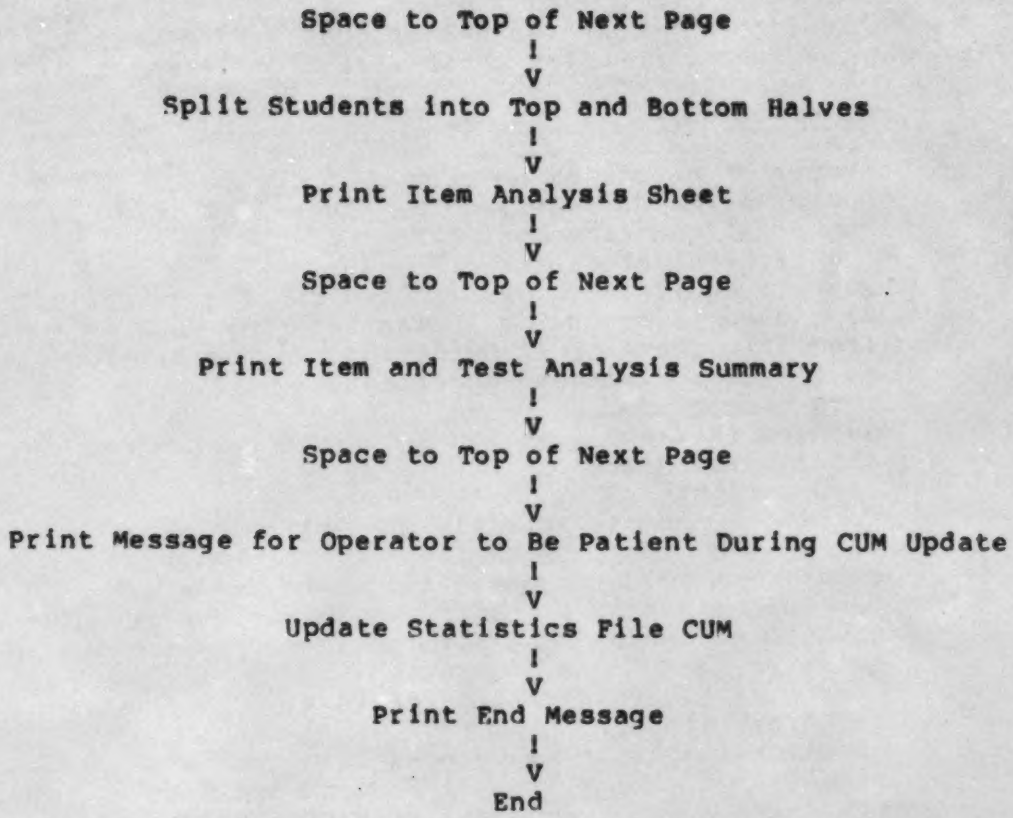


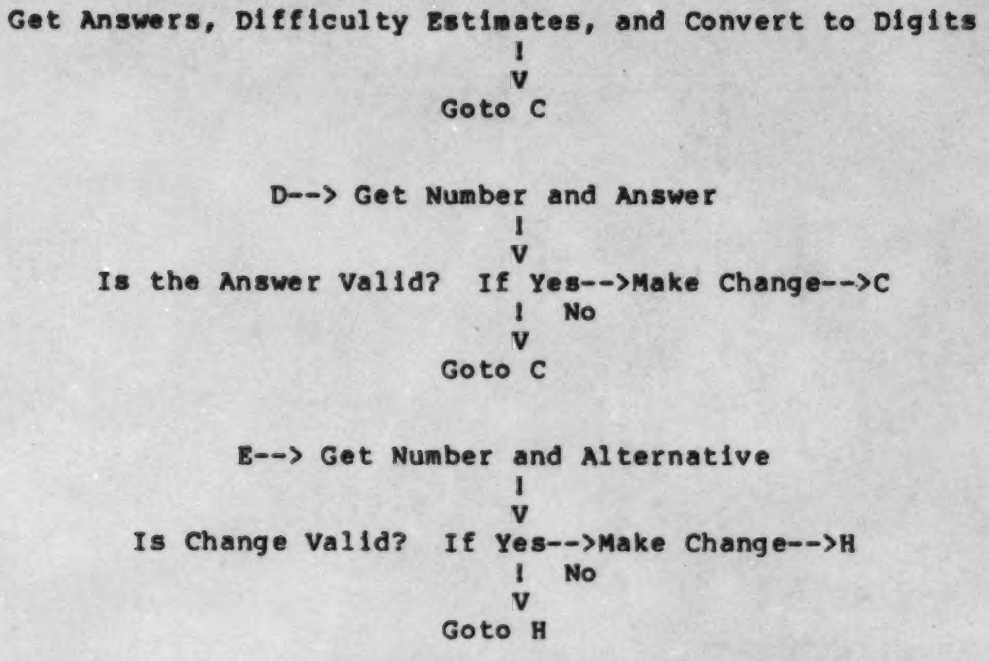
The following list contains all of the numeric variables used in PREP and the functions which those variables serve.

Name	Description
A	number of questions on test
C	number of cards required for student answers
C1	number of questions per card
D(A)	numeric equivalents of all answers
E(A)	numeric equivalents of all item difficulty estimates
F	flag used to determine if test-scoring data is coming from file SDATA or is being input by user; also used as flag for file-copy operations
I	loop parameter
I1,I2	used to indicate question numbers when finding alternative names
J,K	loop parameters
L	number of answers actually present on a given answer card
L1	counter on actual number of student answers
M	maximum number of alternatives in any one question
M1	used to hold number of alternatives in a specific question
N	number of students taking the test
T	temporary storage
V	question number to be changed if answer wrong
W	used in converting item difficulty estimate to numeric equivalent; also used as alternative number when modifying alternative names.

STRING VARIABLES

Name	Description
A$(A,M)	alternative names for each question (string array)
B$	used to hold correct answer to each question
E$	end-of-alternatives mark (***)
G$	used to hold estimate of item difficulty for each question; also used as dummy input parameter
H$	test-heading information
N$	holds one "card's worth" of answers
T$	dummy input parameter


Space to Top of Next Page
Split Students into Top and Bottom Halves
Print Item Analysis Sheet
Space to Top of Next Page
Print Item and Test Analysis Summary
Space to Top of Next Page
Print Message for Operator to Be Patient During CUM Update
Update Statistics File CUM
Print End Message
End


Get Answers, Difficulty Estimates, and Convert to Digits
Goto C
D--> Get Number and Answer
Is the Answer Valid? If Yes-->Make Change-->C
No
Goto C
E--> Get Number and Alternative
Is Change Valid? If Yes-->Make Change-->H
No
Goto H

PREP

```
10 REM *** PROGRAMMED BY VIKTORS MUIZNIEKS 12/77
20 REM *** THIS PROGRAM PREPARES RAW DATA FILE FOR LATER ANALYSIS
30 REM *** BY SCORING PROGRAM
40 REM ***
50 ON ERROR GOTO 1130  'FLAG ERROR IF ONE OCCURS
55 F=0  'INITIALIZE DATA FILE
60 CLEAR 10000  'SET UP STRING STORAGE SPACE
61 PRINT "ENTERING DATA FOR NEW TEST (Y OR N)";
62 G$="": INPUT G$
63 IF G$="Y" THEN 70
64 IF G$<>"N" THEN 61
65 F=1  'INDICATE DATA COMING FROM DATA FILE
66 GOTO 120  'GET HEADING INFO FOR THIS SET OF STUDENTS
70 OPENR "RAW:1"  'PREVIOUSLY LOADED RAW DATA FILE
80 OPENW "DUM:1"  'TEMPORARY STORAGE FILE
90 PRINT "HOW MANY QUESTIONS ON THE TEST";
100 INPUT A
110 IF A<1 THEN 90  'ERROR; MUST HAVE AT LEAST ONE QUESTION
120 PRINT "HEADING INFORMATION (TEACHER NAME-CLASS NAME-PERIOD-DATE)"
130 INPUT H$
135 IF F=1 THEN 2200  'GETTING TEST SPEC.'S FROM FILE?
140 PRINT "MAXIMUM NUMBER OF ALTERNATIVES IN ANY ONE QUESTION";
150 M=0: INPUT M
160 IF M<2 THEN 140
170 REM ***
180 REM *** READ CHANGEABLE PARAMETERS
190 REM *** C CONTAINS # OF QUESTIONS (COLUMNS) PER CARD
200 REM *** E$ CONTAINS END-OF-ALTERNATIVES (EOA) MARK
210 REM ***
220 READ C1,E$
230 DIM A$(A,M),D(A),E(A)  'SET UP STORAGE SPACE FOR ALTERNATIVES
240 I1=1  'STARTING QUESTION NUMBER
250 PRINT "NUMBER OF FIRST QUESTION WHICH HAS DIFFERENT NAMES FOR THE"
260 PRINT "ALTERNATIVES THAN QUESTION NUMBER ";I1;"(0 IF NONE)";
270 INPUT I2
280 IF I2=0 THEN 490  'REST OF QUESTIONS HAVE SAME NUMBER OF ALTERNATIVES
290 IF I2<I1+1 OR I2>A THEN 250  'ERROR; ILLEGAL QUESTION NUMBER
300 PRINT "NUMBER OF ALTERNATIVES IN QUESTION NUMBER ";I1;
310 INPUT M1
320 IF M1<2 OR M1>M THEN 300  'ERROR; ILLEGAL # OF ALTERNATIVES
330 FOR I=1 TO M1
340 PRINT "WHAT IS ALTERNATIVE NUMBER ";I;" CALLED";
350 A$(I1,I)="": INPUT A$(I1,I)
360 IF LEN(A$(I1,I))<>1 THEN 340  'MUST BE ONLY ONE CHARACTER ANSWERS
370 NEXT I
380 IF M1<M THEN A$(I1,M1+1)=E$  'INDICATE END OF ALTERNATIVES
390 IF I2=I1+1 THEN 460  'JUST ONE QUESTION?
400 FOR I=I1+1 TO I2-1  'UPDATE OTHER QUESTION ALTERNATIVE NAMES
410 FOR J=1 TO M1
420 A$(I,J)=A$(I1,J)
430 NEXT J
440 IF M1<M THEN A$(I,M1+1)=E$  'APPEND END OF ALTERNATIVES MARK
450 NEXT I
460 IF I2=A+1 THEN 505  'ALL DONE?
470 I1=I2  'NO, GO BACK FOR REST OF QUESTIONS
480 GOTO 250
490 I2=A+1  'ADD REST OF ALTERNATIVE NAMES
500 GOTO 300
505 GOSUB 4000  'SEE IF WANT TO ADD/CHANGE ANY ALTERNATIVE NAMES
510 PRINT "ENTER CORRECT ANSWER & DIFFICULTY ESTIMATE FOR EACH QUESTION"
520 PRINT "LEGAL DIFFICULTY ESTIMATES ARE H, M, E, AND ?"
530 FOR I=1 TO A
540 PRINT I;
550 B$="":G$="": INPUT B$,G$
560 IF B$=E$ THEN 540  'ERROR; TRIED TO USE RESERVED EOA MARK
570 FOR J=1 TO M
580 IF B$<>A$(I,J) THEN 610  'VALID ANSWER?
590 B=J 'CONVERT ANSWER TO NUMERIC EQUIVALENT
600 GOTO 640  'SEE IF DIFFICULTY ESTIMATE IS LEGAL.
610 IF A$(I,J)=E$ THEN 540 'ANSWER NOT FOUND, ERROR, TRY AGAIN
620 NEXT J
630 GOTO 540 'ANSWER NOT FOUND, ERROR, TRY AGAIN
640 W=4  'DETERMINE IF DIF. EST. IS LEGAL
650 IF G$="H" THEN W=3
660 IF G$="M" THEN W=2
670 IF G$="E" THEN W=1
680 IF G$="?" THEN W=0
690 IF W=4 THEN 540  'NOT LEGAL?
720 D(I)=B  'SAVE NUMERIC EQUIVALENT OF ANSWER
730 E(I)=W 'AND ESTIMATE OF ITEM DIFFICULTY
740 NEXT I     'GET NEXT QUESTION
745 GOSUB 3000  'SEE IF WANT TO CHANGE QUESTION ANSWERS/ESTIMATES OF DIF
750 N=0  'SET COUNTER ON NUMBER OF STUDENTS TO ZERO
760 C=INT(A/C1)+1  'DETERMINE # OF CARDS REQUIRED FOR STUDENT ANSWERS
770 IF INT(A/C1)*C1=A THEN C=C-1 'ADJUST IF HIT CARD BOUNDARY EXACTLY
780 ON ERROR GOTO 1120 'READ ALL STUDENT ANSWERS
790 DSKIN N$ 'GET STUDENT NAME
800 N=N+1 'INCREASE COUNT ON NUMBER OF STUDENTS
810 DSKOUT N$ 'STORE STUDENT NAME
820 L1=0  'COUNTER ON # OF STUDENT ANSWERS
830 FOR I=1 TO C  'FOR ALL CARDS
840 DSKIN N$  'READ IN A CARD'S WORTH OF ANSWERS
850 L=LEN(N$)  'DETERMINE LENGTH OF ANSWER CARD
860 FOR J=1+(I-1)*C1 TO I*C1 'BEWARE MULTIPLE CARDS
```

13

14

PREP CONTINUED

```
870 IF L1>A THEN 1040   'TEST FOR TOO MANY ANSWERS ON CARD
871 IF J=(I-1)*C1<=L THEN 890
872 L1=L1+1   'INCREASE QUESTION ANSWER COUNTER
873 IF L1>A THEN 1040   'SEE IF TOO MANY ANSWERS
874 DSKOUT 0   'IF NOT, ANSWER IS BLANK
875 DSKOUT " "
876 GOTO 1040
880 G$=MID$(N$,J-(I-1)*C1,1)   'GET EACH ANSWER
890 FOR K=1 TO M   'FIND IF IT EXISIS
900 IF J>A THEN 1040   'TEST FOR TOO MANY ANSWERS ON CARD
910 IF L1>A THEN 1040   'TEST FOR TOO MANY ANSWERS ON CARD
920 IF G$<>A$(I,K) THEN 970   'NOT ANSWER?
930 L1=L1+1   'INCREASE COUNTER ON # OF STUDENT ANSWERS
940 IF L1>A THEN 1040   'MORE ANSWERS THAN QUESTIONS?
950 DSKOUT K   'STORE NUMERIC EQUIVALENT OF ANSWER
960 GOTO 1040   'AND DECODE NEXT ANSWER
970 IF A$(I,K)<>E$ THEN 1030   'POSSIBILITIES EXHAUSTED?
980 L1=L1+1   'INCREASE COUNTER ON # OF STUDENT ANSWERS
990 IF L1>A THEN 1040   'MORE ANSWERS THAN QUESTIONS?
1000 DSKOUT 0   'STORE UNDECIPHERABLE ANSWER FLAG
1010 DSKOUT G$   'AND EXACT WRONG ANSWER
1020 GOTO 1040   'AND WORK ON NEXT ANSWER
1030 NEXT K   'GET NEXT ALTERNATIVE TO SAME QUESTION
1031 L1=L1+1   'ALT NOT FOUND
1032 IF L1>A THEN 1040
1033 DSKOUT 0   'STORE UNDECIPHRABLE ANSWER
1034 DSKOUT G$
1040 NEXT J   'GET NEXT ANSWER
1050 NEXT I   'GET NEXT CARD
1060 IF L1>A THEN 780   'TOO FEW ANSWERS?
1070 FOR I=1 TO A-L1   'FOR ALL UNANSWERED QUESTIONS
1080 DSKOUT 0   'OUTPUT BLANK ANSWER
1090 DSKOUT " "
1100 NEXT I
1110 GOTO 780   'GET NEXT STUDENT
1120 IF ERR=27 THEN RESUME 1150
1130 PRINT "ERROR #";ERR;"IN LINE #";ERL
1140 STOP
1150 PRINT "THE PROGRAM IS RECORDING YOUR RESPONSES."
1160 PRINT "PLEASE BE PATIENT."
1170 CLOSR   'CLOSE AND REOPEN PROPER FILES
1180 CLOSW
1190 OPENR "DUM.1"
1200 OPENW "SCORE.1"   'REAL DATA FILE
1210 REM ***
1220 REM *** NEED 2-FILE PROCESS BECAUSE CANNOT GET # OF STUDENTS (N)
1230 REM *** ON FIRST PASS
1240 REM ***
1245 F=0   'INDICATE COPYING INTO SCORE FILE
1250 DSKOUT A   'STORE # OF QUESTIONS
1260 DSKOUT M$   'STORE TEST HEADING INFORMATION
1270 FOR I=1 TO A   'FOR EACH QUESTION
1280 DSKOUT D(I)   'STORE CORRECT ANSWER
1290 DSKOUT E(I)   'AND ESTIMATE OF DIFFICULTY
1320 NEXT I
1330 DSKOUT N   'STORE NUMBER OF STUDENTS TAKING TEST***THE TROUBLEMAKER
1340 DSKOUT M   'STORE MAX # OF ALTERNATIVES USED IN AN ITEM
1350 FOR I=1 TO A   'FOR ALL QUESTIONS
1360 FOR J=1 TO M   'FOR ALL ALTERNATIVES
1370 DSKOUT A$(I,J)   'STORE ACTUAL ALTERNATIVE NAMES
1380 IF A$(I,J)=E$ THEN 1400   'TEST FOR END OF ALTERNATIVES
1390 NEXT J   'NEXT ALTERNATIVE
1400 NEXT I   'NEXT QUESTION
1405 IF F=1 THEN 2140   'SEE IF COPYING INTO SDATA FILE
1410 FOR I=1 TO N   'FOR ALL STUDENTS
1420 DSKIN T$   'STORE STUDENT NAME
1430 DSKOUT T$
1440 FOR I=1 TO A   'FOR ALL QUESTIONS
1450 DSKIN T
1460 DSKOUT T   'STORE STUDENT ANSWERS
1470 IF T<0 THEN 1500   'CHECK FOR PRESENCE OF UNRECOGNIZABLE ANSWER
1480 DSKIN T$
1490 DSKOUT T$   'STORE UNRECOGNIZABLE ANSWER IF PRESENT
1500 NEXT J   'NEXT QUESTION
1510 NEXT I   'NEXT STUDENT
1520 CLOSR   'CLOSE ALL FILES
1530 CLOSW
1531 REM *** FIND OUT IF WANT DATA STORED FOR ANOTHER CLASS
1532 PRINT "STORE DATA FOR ANOTHER CLASS (Y OR N)";
1533 G$="":  INPUT G$
1534 IF G$="N" THEN 1540
1535 IF G$="Y" THEN 2100
1536 GOTO 1532
1540 REM ***
1550 REM *** COMMENT TO CARD GRADING SYSTEM OPERATOR/TEACHER
1560 REM ***
1570 PRINT "TYPE IN EDOS AND PRESS THE RETURN KEY AFTER OK IS PRINTED."
1575 GOTO 9000   'ALL DONE
1580 REM ***
1590 REM *** DATA VALUES  CI, E$
1600 REM ***
1610 DATA 40
1620 DATA "***"
2100 REM *** COPY DATA INTO STORAGE FILE SDATA
```

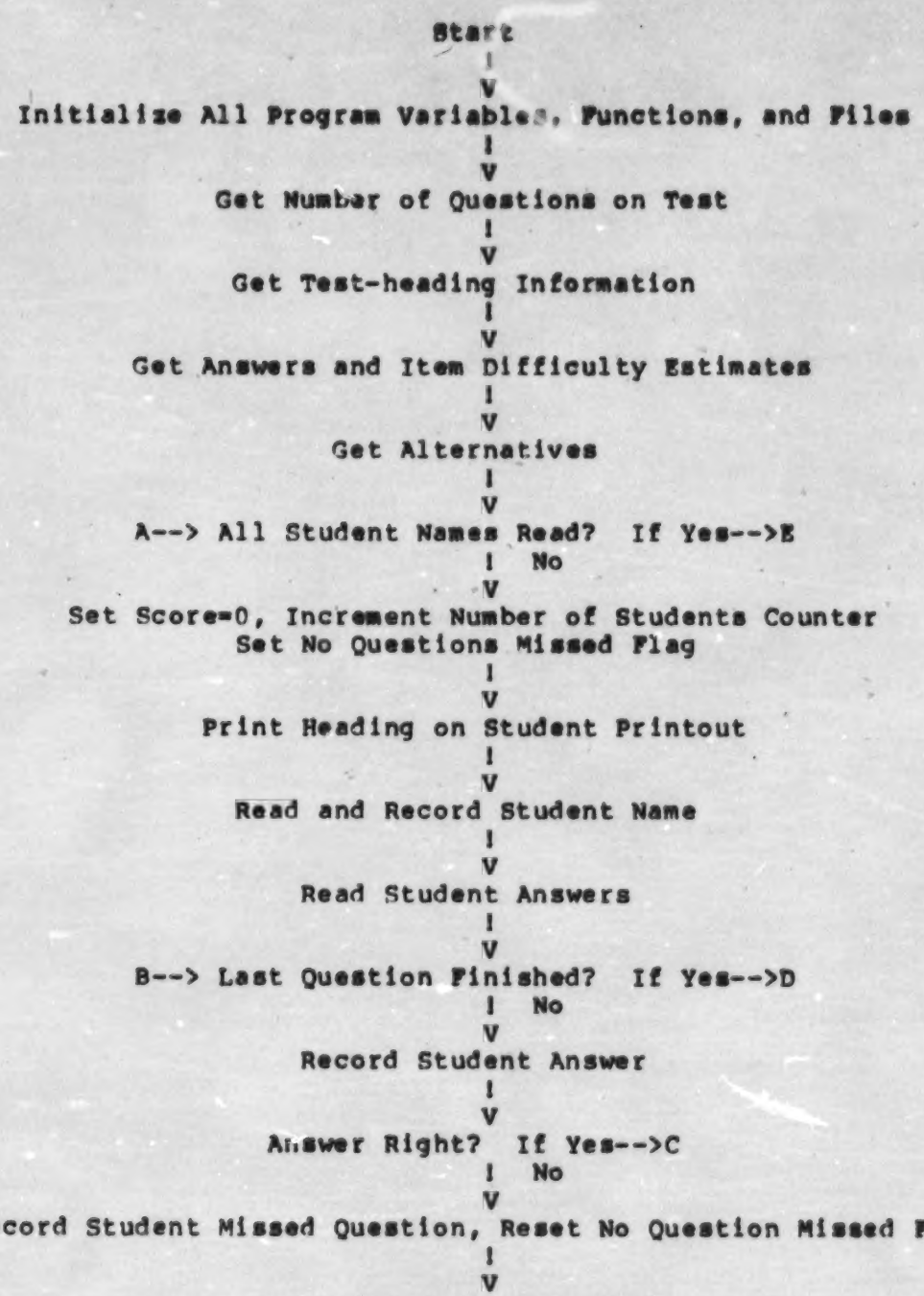
PREP CONTINUED

```
2105 OPENR "SCORE:1"
2110 OPENW "SDATA:1"
2120 F=1  'INDICATE WRITING INTO FILE SDATA
2130 GOTO 1250  'AND COPY INTO PROPER FILE
2140 CLOSR
2150 CLOSW
2160 GOTO 1570  'EXIT IN NORMAL FASHION
2190 REM *** READ BACK DATA VALUES FORM SDATA FILE
2200 OPENR "SDATA.1"
2210 DSKIN A  'GET # OF QUESTIONS ON TEST
2220 DSKIN G$  'IGNORE OLD HEADING INFO
2230 READ C1,E$  'READ CHANGEABLE PARAMETERS
2235 DIM D(A),E(A)  'RESERVE COR. ANS. AND DIF. SPACE
2240 FOR I=1 TO A
2250 DSKIN D(I)  'GET CORRECT ANSWER
2260 DSKIN E(I)  'GET ESTIMATE OF ITEM DIFFICULTY
2270 NEXT I
2271 DSKIN N  'GET OLD NUMBER OF STUDENTS
2272 DSKIN M  'GET NUMBER OF ALTERNATIVES
2273 DIM A$(A,M)  'ALTERNATIVE NAMES
2280 FOR I=1 TO A
2290 FOR J=1 TO M
2300 DSKIN A$(I,J)  'GET ALT. NAMES
2305 IF A$(I,J)=E$ THEN 2320  'SEE IF FINISHED WITH QUESTION
2310 NEXT J
2320 NEXT I
2321 CLOSR
2322 OPENR "RAW:1"  'SET UP FILES AS NORMAL
2323 OPENW "DUM:1"
2330 GOTO 745  'RETURN TO PROCESSING AS IF ANSWERS FINISHED
3000 REM *** CHANGE ANY CORRECT ANSWERS AND/OR DIF  GUESSES
3010 PRINT "CHANGE ANY ANSWERS OR ESTIMATES (Y OR N)";
3020 G$="": INPUT G$
3030 IF G$="N" THEN RETURN
3040 IF G$<>"Y" THEN 3010
3050 PRINT "WHICH QUESTION DO YOU WANT TO CHANGE (TYPE 0 FOR NONE)";
3060 V=-1: INPUT V
3070 IF V=0 THEN RETURN
3080 IF V<1 OR V>A THEN 3050  'ERROR; OUT OF LEGAL RANGE
3090 PRINT "CORRECT ANSWER AND DIFFICULTY ESTIMATE FOR QUESTION";V;
3100 B$="": G$="": INPUT B$,G$
3110 IF B$=E$ THEN 3090  'ERROR; USING EOA MARK
```

PREP CONTINUED

```
3120 FOR J=1 TO M
3130 IF B$<>A$(V,J) THEN 3160  'ANSWER CORRECT?
3140 B=J  'CONVERT TO NUMERIC EQUIVALENT
3150 GOTO 3190
3160 IF A$(V,J)=E$ THEN 3090  'ALL ANSWERS CHECKED
3170 NEXT J
3180 GOTO 3090  'ANSWER NOT FOUND
3190 W=4  'CHECK ON LEGALITY OF DIFFICULTY ESTIMATE
3195 IF G$="H" THEN W=3
3200 IF G$="M" THEN W=2
3210 IF G$="E" THEN W=1
3220 IF G$="?" THEN W=0
3230 IF W=4 THEN 3090  'ESTIMATE NOT LEGAL?
3240 B(V)=B  'STORE NEW CORRECT ANSWER
3250 F(V)=W  'AND NEW DIFFICULTY ESTIMATE
3260 GOTO 3000  'SEE IF WANT TO ALTER ANOTHER
4000 REM *** ADD/CHANGE QUESTION ALTERNATIVE NAMES
4010 PRINT "ADD OR CHANGE ANY ALTERNATIVE NAMES (Y OR N)";
4020 G$="": INPUT G$
4030 IF G$="N" THEN RETURN
4040 IF G$<>"Y" THEN 4010
4050 PRINT "QUESTION NUMBER (TYPE 0 FOR NONE)";
4060 V=-1: INPUT V
4070 IF V=0 THEN RETURN
4080 IF V<1 OR V>A THEN 4050  'ERROR: ILLEGAL QUESTION NUMBER
4090 PRINT "ALTERNATIVE NUMBER (TYPE 0 FOR NONE)";
4100 W=-1: INPUT W
4110 IF W=0 THEN 4050  'TRY ANOTHER QUESTION
4120 IF W<1 OR W>M THEN 4090  'ERROR: ILLEGAL ALTERNATIVE NUMBER
4130 PRINT "NEW NAME FOR ALTERNATIVE #";W;"IN QUESTION #";V;
4140 G$="": INPUT G$
4145 IF LEN(G$)<>1 THEN 4130  'MUST BE ONLY ONE CHARACTER ANSWERS
4150 FOR J=1 TO M  'DETERMINE WHERE EOA MARK IS IN ANSWER
4160 IF A$(V,J)=E$ THEN 4190  'FOUND IT
4170 NEXT J
4180 J=M+1  'EOA MARK NOT PRESENT
4190 IF W<J THEN 4240  'CHANGING EXISTING NAME
4200 IF W=J THEN 4235  'ADDING NEW ALTERNATIVE NAME
4210 PRINT "ERROR***  YOU MUST ADD ALTERNATIVE NUMBER";J;"BEFORE YOU"
4220 PRINT "CAN ADD ALTERNATIVE NUMBER";W;"IN QUESTION";V
4230 GOTO 490  'TRIED TO SKIP AN ALTERNATIVE
4235 IF J<M THEN A$(V,J+1)=E$  'ADD NEW EOA MARK IF NECESSARY
4240 A$(V,W)=G$  'ADD NEW ALTERNATIVE NAME
4250 GOTO 4010  'SEE IF WANT TO CHANGE/ADD SOME MORE ALT. NAMES
9000 END
```

The following flowchart is intended as a guide to the logic used in the GRADE program.


Start
Initialize All Program Variables, Functions, and Files
Get Number of Questions on Test
Get Test-heading Information
Get Answers and Item Difficulty Estimates
Get Alternatives
A--> All Student Names Read? If Yes-->E
No
Set Score=0, Increment Number of Students Counter
Set No Questions Missed Flag
Print Heading on Student Printout
Read and Record Student Name
Read Student Answers
B--> Last Question Finished? If Yes-->D
No
Record Student Answer
Answer Right? If Yes-->C
No
Record Student Missed Question, Reset No Question Missed Flag

(continued on next page)

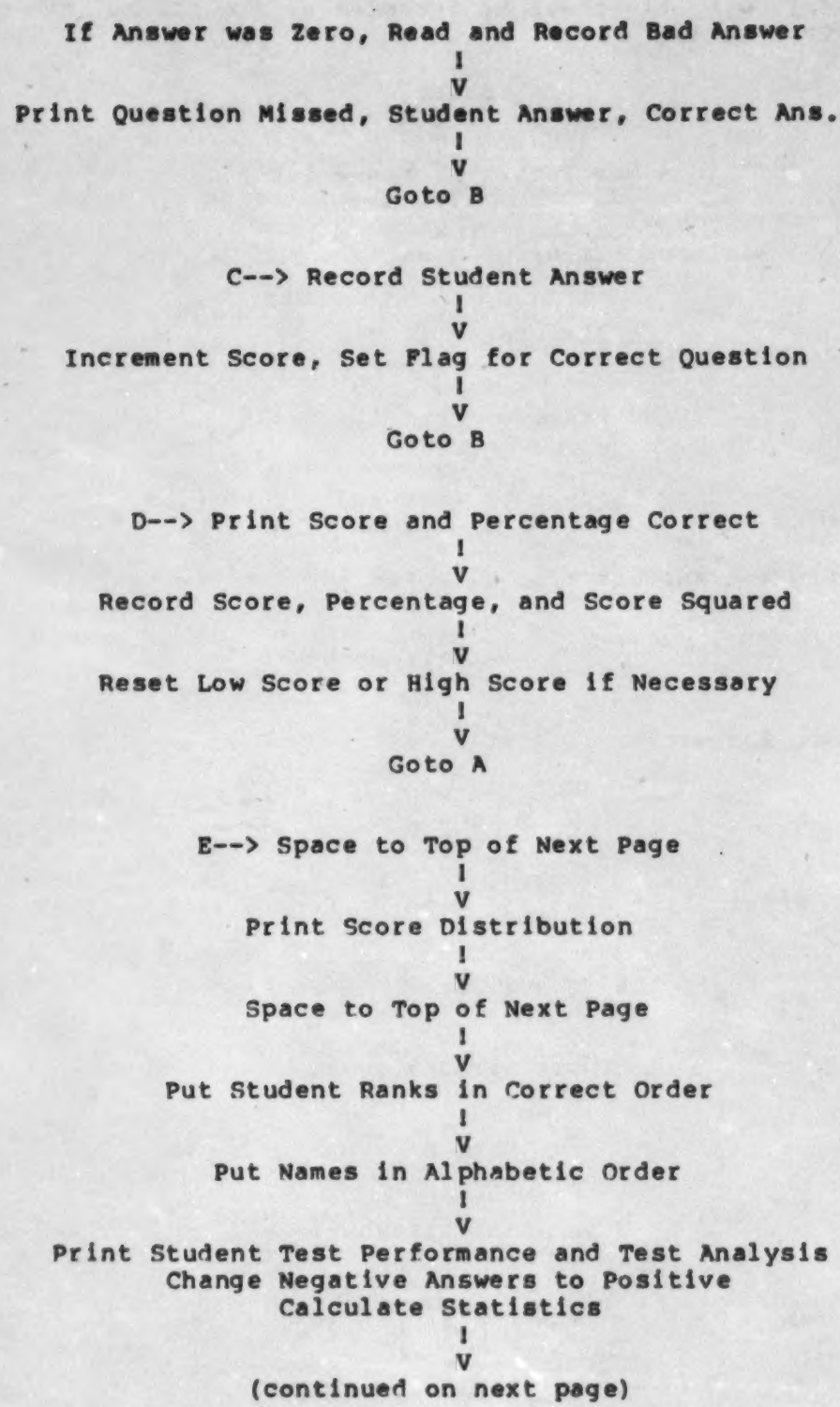
T	temporary storage variable
T6	loop parameter
T7	loop parameter
T8	temporary storage variable
U9	highest rank of people in top-half of the class
V%(A,M)	number of students choosing each alternative for all questions
W1	standard error of measurement
W2	item discriminating power
W3	sum of all item difficulty powers
W8	score one standard deviation above mean; also used in calculating item efficiency
W9	used to adjust test reliability to 100 item test via Spearman-Brown prophecy formula; also used as temporary storage in calculating average score
X	sum of all (raw scores squared)
Y6	number of questions with good discrimination
Y7	number of questions with marginal discrimination
Y8	number of questions with exceptionally good discrimination
Y9	number of questions which do not discriminate
Z5	number of unchosen distractors in easy questions
Z7	number of possibly ambiguous questions
Z8	number of unchosen distractors in medium questions
Z9	number of unchosen distractors in hard questions

STRING VARIABLES

Name	Description
A$(A,M)	alternative names for all questions
E1$	end-of-alternatives mark (***)
G$(3)	names of item difficulty types
H$	test-heading information
N$	temporary storage variable
N$(N)	student names
Q$	dummy input parameter; also used in printing student names

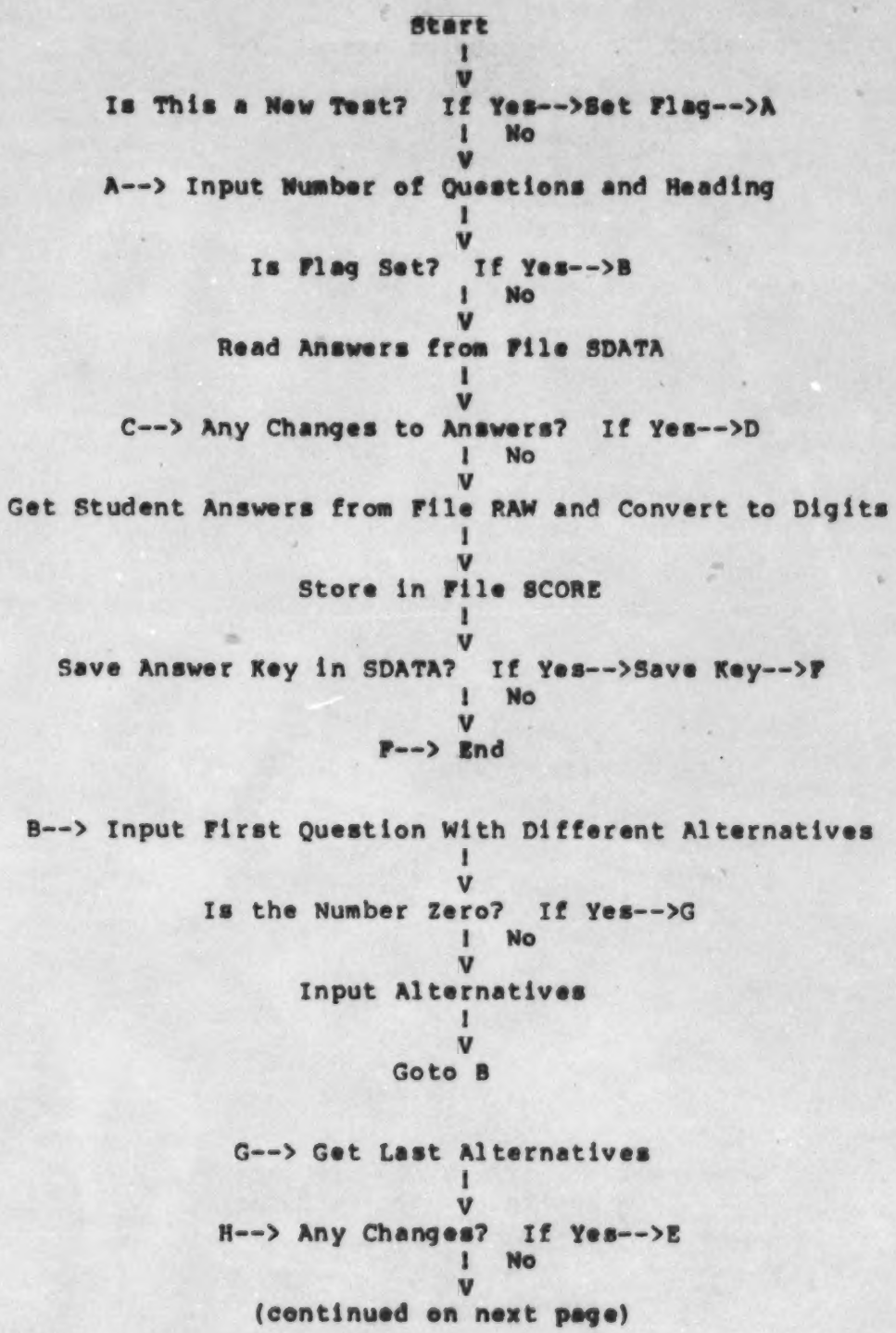
The following list contains all of the numeric variables used in GRADE and the functions which those variables serve. (A % sign indicates that the variable can only assume integer values.)

Name	Description
A	number of questions on test
B%(A)	correct answer to each question
C1	score one standard deviation below mean
C2	standard deviation
C3	sum of proportion passing each item
C4	Kuder-Richardson formula 20 test reliability value
ERR	system variable which holds the latest type of error to occur
ERL	system variable which holds the line number at which the latest error occurred
F	flag used to indicate student has no wrong answers; also used as flag in file-copy operations
F7	counter for number of questions printed per line on student printouts, and number of alternatives printed per line on item analysis printout
F8	flag used in printing questions missed by each student
G%(A)	estimate of item difficulty for each question
I,J	loop parameters
K	counter used in alphabetizing student names
K1	used as index into K2 array
K2%(3)	counters on number of easy, medium, and hard questions
K6	number of ? distractors
K7	number of + distractors
K8	total number of distractors
K9	number of unchosen distractors
M	maximum number of alternatives to any one question
M1	current high score
M2	current low score
M3	counter on number of lines printed
M4	number of lines that can be printed on a page
N	number of students
R%(N)	raw scores for all students
R1%(N)	rank of each student
R2%(N)	index values for achieving alphabetic order in student names
R3	number of students (in top-half of class) getting correct answer to a question
R4	number of students (in bottom-half of class) getting correct answer to a question
R5	distractor for which discriminating power being calculated
R5%(N)	index values relating to alphabetization of student names
S	sum of all raw scores
S2%(N,A)	all student responses to each question
S9	number of correct item difficulty estimates


If Answer was Zero, Read and Record Bad Answer
Print Question Missed, Student Answer, Correct Ans.
Goto B
C--> Record Student Answer
Increment Score, Set Flag for Correct Question
Goto B
D--> Print Score and Percentage Correct
Record Score, Percentage, and Score Squared
Reset Low Score or High Score if Necessary
Goto A
E--> Space to Top of Next Page
Print Score Distribution
Space to Top of Next Page
Put Student Ranks in Correct Order
Put Names in Alphabetic Order
Print Student Test Performance and Test Analysis
Change Negative Answers to Positive
Calculate Statistics

(continued on next page)

The following flowchart is intended as a guide to the logic of the PREP program.


Start
Is This a New Test? If Yes-->Set Flag-->A
No
A--> Input Number of Questions and Heading
Is Flag Set? If Yes-->B
No
Read Answers from File SDATA
C--> Any Changes to Answers? If Yes-->D
No
Get Student Answers from File RAW and Convert to Digits
Store in File SCORE
Save Answer Key in SDATA? If Yes-->Save Key-->F
No
F--> End
B--> Input First Question With Different Alternatives
Is the Number Zero? If Yes-->G
No
Input Alternatives
Goto B
G--> Get Last Alternatives
H--> Any Changes? If Yes-->E
No

(continued on next page)

GRADE

```
10 DEFINT A-B
20 DEFINT D-V
30 DEFINT X-Z
40 DEFSNG C
50 DEFSNG W
60 ON ERROR GOTO 4430
70 CLEAR
80 CLEAR 1:000
90 WIDTH 132
100 DEF FNA(X,A)=INT(X*10000/A +  5)/100
110 DEF FNB(X,A)=INT(X*100/A +  5)/100
120 DEF FNC(X,A)=INT(INT(X*10000/A + .5)/100 +  5)
130 READ M4,E1$
140 OPENR "SCORE 1"
150 DSKIN A
160 DSKIN H$
170 DIM B%(A),G%(A)
180 DIM G$(3),K2%(3)
190 G$(0)="     "
200 G$(1)="EASY"
210 G$(2)="MEDIUM"
220 G$(3)="HARD"
230 M2=A
240 FOR I=1 TO A
250 DSKIN B%(I)
260 DSKIN G%(I)
280 NEXT I
290 DSKIN N
300 DIM N$(N),R%(N),R1%(N),R2%(N),R6%(N)
310 DSKIN M
320 DIM A$(A,M),V%(A,M),S2%(N,A)
330 FOR I=1 TO A
340 FOR J=1 TO M
350 DSKIN A$(I,J)
360 IF A$(I,J)=E1$ THEN 380
370 NEXT J
380 NEXT I
390 FOR I=1 TO N
400 DSKIN N$(I)
420 LPRINT
430 LPRINT
440 LPRINT
450 LPRINT H$
460 LPRINT
470 F7=0
480 LPRINT N$(I)
490 LPRINT
500 R%(I)=0
510 F=0
520 M3=M3+2
530 FOR J=1 TO A
540 DSKIN I
550 V%(J,I)=V%(J,I)+1
560 S2%(I,J)=1
570 IF I=B%(J) THEN 780
580 S2%(I,J)=-S2%(I,J)
600 IF F<>0 THEN 660
610 LPRINT "YOU MISSED THE FOLLOWING QUESTION(S).  YOUR ANSWER IS ";
620 LPRINT "PRINTED FIRST, FOLLOWED BY THE CORRECT ANSWER."
630 LPRINT
640 F=1
650 M3=M3+2
660 IF F<>0 THEN 710
670 O$="": DSKIN O$
680 IF O$="" THEN O$=" "
690 LPRINT STR$(J);"  (";O$;")--->(";A$(J,B%(J));")    ";
700 GOTO 720
710 LPRINT STR$(J);"  (";A$(J,I);")--->(";A$(J,B%(J));")    ";
720 F7=F7+1
730 IF F7<7 THEN 790
740 F7=0
750 LPRINT
760 M3=M3+1
770 GOTO 790
780 R%(I)=R%(I)+1
790 NEXT J
810 LPRINT
820 M3=M3+1
830 S=S+R%(I)
850 IF F=0 THEN LPRINT "NO QUESTIONS MISSED"
860 LPRINT
870 LPRINT "SCORE = ";R%(I);" OUT OF ";A
880 LPRINT
890 LPRINT "PERCENTAGE CORRECT = ";FNB(R%(I),A)*100
900 IF F=0 THEN M3=M3+5 ELSE M3=M3+4
910 IF R%(I)<M2 THEN M2=R%(I)
920 IF R%(I)>M1 THEN M1=R%(I)
930 X=X+R%(I)*R%(I)
940 NEXT I
950 GOTO 1140
```

GRADE CONTINUED

```
960 FOR T6=1 TO N
970 T8=T6
980 FOR T7=1 TO N
990 IF R1%(T7)<>T6 THEN 1030
1000 IF T8=T6 THEN 1020
1010 R1%(T7)=T8
1020 T8=T8+1
1030 NEXT T7
1040 NEXT T6
1050 RETURN
1060 IF M3=M4 THEN 1130
1070 IF M3<M4 THEN 1100
1080 M3=M3-M4
1090 GOTO 1060
1100 FOR I=1 TO M4-M3
1110 LPRINT
1120 NEXT I
1130 RETURN
1140 GOSUB 1060
1145 GOSUB 4740
1150 LPRINT
1160 LPRINT
1170 LPRINT H$
1180 LPRINT
1190 LPRINT
1200 LPRINT "STUDENT NAME";TAB(25);"SCORE";TAB(35);"PERCENTAGE";
1210 LPRINT TAB(50);"RANK";TAB(57);"GRADE";TAB(66);"QUESTION(S) MISSED"
1220 LPRINT
1230 M3=7
1240 GOSUB 2420
1250 GOSUB 2550
1260 FOR I=1 TO N
1275 Q$=MID$(N$(R2%(I)),1,23)
1280 LPRINT Q$;TAB(26);R%(R2%(I));TAB(37);FNB(R%(R2%(I)),A)*100;
1290 LPRINT "%";TAB(50);R1%(R2%(I));TAB(57);"(  )";TAB(66);
1291 F8=0
1292 FOR J=1 TO A
1293 IF S2%(R2%(I),J)>0 THEN 1307
1294 S2%(R2%(I),J)=-S2%(R2%(I),J)
1295 F8=F8+1
1296 IF F8=1 THEN 1300
1297 IF F8=21 THEN 1302
1298 LPRINT ",";MID$(STR$(J),2);
1299 GOTO 1307
1300 LPRINT MID$(STR$(J),2);
1301 GOTO 1307
1302 LPRINT
1303 M3=M3+1
1304 LPRINT TAB(66);
1305 F8=1
1306 GOTO 1300
1307 NEXT J
1308 LPRINT:M3=M3+1
1309 NEXT I
1310 LPRINT
1320 LPRINT
1330 LPRINT "THE TEST SCORES RANGED FROM";M2;"TO";M1;"--- A SPREAD OF";
1340 LPRINT M1-M2;"POINTS."
1350 LPRINT
1360 W9=FNB(S,N)
1370 LPRINT "THE AVERAGE SCORE IS";INT(W9+.5);"POINTS."
1380 LPRINT
1390 C2=FNB(SQR(N*X-S*S),N)
1400 W8=INT(W9+C2+.5)
1410 C1=INT(W9-C2)
1420 IF W8>A THEN W8=A
1430 IF C1<0 THEN C1=0
1440 C3=0
1450 FOR I=1 TO A
1460 C3=C3+((N-V%(I,B%(I)))/N)*(V%(I,B%(I))/N)
1470 NEXT I
1480 C4=0
1490 IF C2=0 THEN 1530
1500 C4=A/(A-1)*((C2*C2-C3)/(C2*C2))
1504 IF C4>1 THEN C4=1
1505 IF C4>0 THEN 1510
1506 C4=0
1507 GOTO 1530
1510 W9=100/A
1520 C4=FNB(W9*C4,(1+(W9-1)*C4))
1530 LPRINT "THE THEORETICAL VALUE OF YOUR TEST RELIABILITY =";C4;
1540 LPRINT"(OUT OF A POSSIBLE 1.00)   (THE AVERAGE RELIABILITY OF ";
1544 LPRINT "TEACHER-MADE TESTS IS"
1550 LPRINT "ABOUT  .60).  THEREFORE, YOU ";
1555 LPRINT "PROBABLY CAN ASSUME THAT THE RESULTS OF YOUR TEST ARE ";
1560 IF C4<.4 THEN LPRINT "SOMEWHAT QUESTIONABLE."
1565 IF C4<.75 AND C4>=.4 THEN LPRINT "RELIABLE."
1570 IF C4>=.75 THEN LPRINT "VERY RELIABLE."
1575 LPRINT "BEWARE!!!   A GOOD TEST SHOULD BE RELIABLE.  HOWEVER, ";
1580 LPRINT "JUST BECAUSE A TEST IS RELIABLE DOESN'T MAKE IT GOOD!"
```

GRADE CONTINUED

```
1585 LPRINT "A VERY RELIABLE TEST COULD BE TESTING SOMETHING DIFFERENT";
1586 LPRINT " FROM WHAT YOU WANT TO TEST."
1590 LPRINT
1600 LPRINT "THE SCORES ON THE TEST HAVE A STANDARD DEVIATION OF";C2;
1610 LPRINT "(THIS MEANS THAT ABOUT TWO-THIRDS OF THE SCORES FALL ";
1620 LPRINT "BETWEEN";C1;"AND";W8;") "
1630 LPRINT"FROM THE STANDARD DEVIATION, AND THE TEST RELIABILITY, IT ";
1640 LPRINT"IS THEORETICALLY POSSIBLE TO CALCULATE HOW MANY POINTS EACH"
1650 LPRINT"STUDENT MAY HAVE MISSED (OR GOTTEN RIGHT) BY CHANCE FACTOR";
1660 LPRINT "S ALONE (SUCH AS GUESSING)   FOR THIS TEST, YOU CAN ASSUME"
1670 LPRINT "THAT EACH STUDENT'S SCORE IS CORRECT TO WITHIN";
1680 W1=FNR(C2*SQR(1-C4),1)
1690 LPRINT INT(W1+.5);"POINT(S)."
1700 M3=M3+15
1710 GOSUB 1060
1720 LPRINT
1730 LPRINT H$
1740 LPRINT
1750 LPRINT
1760 LPRINT "QUESTION";TAB(10);"STUDENT RESPONSE DISTRIBUTION";
1770 LPRINT TAB(52);"% CORRECT ANS ";
1780 LPRINT TAB(67),"  DIFFICULTY      ESTIMATE OF QUESTION ";
1790 LPRINT "DISCRIMINATION ABILITY"
1800 LPRINT
1810 M3=6
1820 U9=INT(N/2)
1830 L9=INT(N/2+1.8)
1840 GOSUB 960
1850 FOR I=1 TO A
1860 LPRINT TAB(2);I;
1870 F7=0
1880 W9=FNR(V%(I,B%(I)),N)
1890 FOR J=1 TO M
1900 IF A$(I,J)=F1$ THEN 2100
1910 F7=F7+1
1920 IF F7<7 THEN 1960
1930 F7=1
1940 LPRINT
1950 M3=M3+1
1960 IF B%(I)=J THEN 2080
1970 LPRINT TAB(F7*7+3);A$(I,J);"=";STR$(V%(I,J));
1980 IF V%(I,J)=0 THEN GOSUB 2900
1990 K8=K8+1
2000 IF V%(I,J)<2 THEN 2090
2010 R5=J
2020 GOSUB 2270
2030 IF W2>0 THEN LPRINT "?";
2040 IF W2>0 THEN K6=K6+1
2050 IF W2<= 5 THEN LPRINT "+";
2060 IF W2<= 5 THEN K7=K7+1
2070 GOTO 2090
2080 LPRINT TAB(F7*7+2);"*";A$(I,J);"=";STR$(V%(I,J));
2090 NEXT J
2100 IF V%(I,0)=0 THEN 2170
2110 IF F7<5 THEN 2160
2120 LPRINT
2130 M3=M3+1
2140 LPRINT TAB(10);"OMIT =";V%(I,0);
2150 GOTO 2170
2160 LPRINT "  OMIT =";V%(I,0);
2170 IF W9> 25 THEN 2200
2180 K1=3
2190 GOTO 2210
2200 IF W9> 75 THEN K1=1 ELSE K1=2
2210 K2%(K1)=K2%(K1)+1
2220 LPRINT TAB(56);W9*100;"%";
2224 T6=69
2225 IF G%(I)<30 THEN 2230
2226 GOTO 2240
2230 LPRINT TAB(T6);G$(G%(I));
2235 T6=76
2240 W3=W3+W9
2250 IF K1<>G%(I) THEN LPRINT TAB(T6);"(";G$(K1);")"; ELSE S9=S9+1
2260 R5=B%(I)
2270 R3=0
2280 R4=0
2290 FOR T6=1 TO N
2300 IF R1%(T6)>U9 THEN 2330
2310 IF S2%(T6,1)=R5 THEN R3=R3+1
2320 GOTO 2350
2330 IF R1%(T6)<L9 THEN 2350
2340 IF S2%(T6,1)=R5 THEN R4=R4+1
2350 NEXT T6
2360 W2=(R3-R4)/U9
2370 IF R5<>B%(I) THEN RETURN
2380 GOSUB 2660
2390 NEXT I
2400 M3=M3+A
2410 GOTO 3020
2420 R1%(1)=1
```

GRADE CONTINUED

```
2430 FOR I=2 TO N
2440 R1%(I)=I
2450 F6=0
2460 FOR J=1 TO I-1
2470 IF R%(I)=R%(J) THEN F6=R1%(J)
2480 IF R%(I)<=R%(J) THEN 2510
2490 IF R1%(J)<R1%(I) THEN R1%(I)=R1%(J)
2500 R1%(J)=R1%(J)+1
2510 NEXT J
2520 IF F6<>0 THEN R1%(I)=F6
2530 NEXT I
2540 RETURN
2550 FOR I=1 TO N
2560 K=0
2570 FOR J=1 TO N
2580 IF N$(I)>N$(J) THEN K=K+1
2590 NEXT J
2600 R6%(I)=K+1
2610 NEXT I
2620 FOR I=1 TO N
2630 R2%(R6%(I))=1
2640 NEXT I
2650 RETURN
2660 LPRINT TAB(85);
2670 IF W2<>0 THEN 2710
2680 LPRINT "QUESTION DOES NOT DISCRIMINATE AMONG STUDENTS"
2690 Y9=Y9+1
2700 RETURN
2710 IF W2>0 THEN 2750
2720 LPRINT "QUESTION DISCRIMINATES THE 'WRONG' WAY"
2730 Z7=Z7+1
2740 RETURN
2750 IF W9> .5 THEN 2780
2760 W8=W2/(2*W9)
2770 GOTO 2790
2780 W8=W2/(2*(1-W9))
2790 IF W8< 0 THEN 2830
2800 LPRINT "EXCEPTIONALLY GOOD DISCRIMINATION"
2810 Y8=Y8+1
2820 RETURN
2830 IF W8< .5 THEN 2870
2840 LPRINT "GOOD DISCRIMINATION"
2850 Y6=Y6+1
2860 RETURN
2870 LPRINT "MARGINAL DISCRIMINATION"
2880 Y7=Y7+1
2890 RETURN
```

GRADE CONTINUED

```
2900 K9=K9+1
2910 IF W9> 25 THEN 2940
2920 Z9=Z9+1
2930 RETURN
2940 IF H9> 75 THEN Z5=Z5+1 ELSE Z8=Z8+1
2950 RETURN
3020 CLOSE:OPENR"TEXT 1":J=1:ERASER%,R1%,R2%,R6%,V%
3021 DIM R%(26):R%(1)=K7:R%(2)=FNC(K7,K8):R%(3)=K6:R%(4)=FNC(K6,K8)
3022 R%(5)=K9:R%(6)=FNC(K9,KH):R%(7)=Z5:R%(8)=Z8:R%(9)=Z9
3023 R%(10)=FNC(S9,A):R%(11)=K2%(1):R%(12)=FNB(K2%(1),A)*100
3024 R%(14)=FNB(K2%(2),A)*100:R%(15)=K2%(3):R%(16)=FNB(K2%(3),A)*100
3025 R%(18)=FNB(Y8,A)*100:R%(19)=Y6:R%(20)=FNB(Y6,A)*100:R%(21)=Y7
3026 R%(22)=FNB(Y7,A)*100:R%(23)=Y9:R%(24)=FNB(Y9,A)*100:R%(25)=Z7
3027 R%(26)=FNB(Z7,A)*100:R%(17)=Y8:R%(13)=K2%(2)
3040 ON ERROR GOTO 3200
3050 Q$="":DSKIN Q$:IF Q$="NUM"THEN3090
3055 IF Q$<>"PAGE"THEN3060ELSEGOSUB1060:LPRINTH$:M3=0:GOTO3040
3060 IF RIGHT$(Q$,1)="#" THEN 3110
3070 LPRINT Q$:M3=M3+1:GOTO3040
3090 LPRINT R%(J);:J=J+1:GOTO3040
3110 A=LEN(Q$)-1:Q$=LEFT$(Q$,A):LPRINT Q$;:GOTO3040
3200 IF ERR=27 THENRESUME4270ELSE4430
4270 GOSUB 1060
4280 CLOSE
4290 PRINT "TYPE IN FDOS AND PRESS THE RETURN KEY AFTER OK IS PRINTED."
4300 PRINT "IT MAY TAKE A LITTLE WHILE. SO PLEASE BE PATIENT"
4310 F=0
4320 OPENR "CUM 1"
4330 OPENW "NUM 1"
4340 ON ERROR GOTO 4420
4350 DSKIN N$
4360 DSKOUT N$
4370 FOR I=1 TO 13
4380 DSKIN T6
4390 DSKOUT T6
4400 NEXT I
4410 GOTO 4340
4420 IF ERR=27 THEN RESUME 4450
4430 PRINT "ERROR #";ERR,"IN LINE #";ERL
4440 STOP
4450 IF F=1 THEN 4650
4460 DSKOUT H$
4470 DSKOUT C4
4480 DSKOUT C2
```

GRADE CONTINUED

```
4490 DSKOUT W1
4500 DSKOUT R%(2)
4510 DSKOUT R%(4)
4520 DSKOUT R%(6)
4580 DSKOUT 100-FNA(S9,A)
4590 DSKOUT FNB(W3,A)
4600 DSKOUT FNA(Y8,A)
4610 DSKOUT FNA(Y6,A)
4620 DSKOUT FNA(Y7,A)
4630 DSKOUT FNA(Y9,A)
4640 DSKOUT FNA(Z7,A)
4650 CLOSR
4660 CLOSW
4670 IF F=1 THEN 4860
4680 OPENR "DUM:1"
4690 OPENW "CUM:1"
4700 F=1
4710 GOTO 4340
4720 DATA 66
4730 DATA "***"
4740 LPRINT:LPRINT H$:LPRINT:LPRINT:LPRINT "SCORE   NUMBER OF STUDE";
4750 LPRINT "NTS WITH THAT SCORE  (EACH * REPRESENTS A STUDENT)":LPRINT
4760 M3=6
4770 FOR I=0 TO A
4780 LPRINT I;TAB(8);
4790 FOR J=1 TO N
4800 IF R%(J)<>I THEN 4820
4810 LPRINT "*";
4820 NEXT J
4830 LPRINT:M3=M3+1
4840 NEXT I
4850 GOSUB 1060
4855 RETURN
4860 END
```

The following list contains all of the variables used in DUMP1 and the functions which those variables serve.

Name	Description
A(13)	computes sum for each statistic stored in CUM
B(13)	stores the minimum value for each statistic
C	used to input the choice of options
C(13)	stores the maximum for each statistic
D(13)	used to calculate standard deviation for each statistic
F	flag to indicate the current test is to be included in the printout
H$(13)	stores the name of each statistic
I	loop parameter
K	used for numeric disk inputs
M$	used to input the teacher's name
N$	used to input character data from disk
S	stores the number of tests included in the printout

APPENDIX C

PROGRAM LISTINGS, FLOW DIAGRAMS, AND DESCRIPTION OF VARIABLE FUNCTIONS

DUMP1

```
1 REM *** PROGRAMMED BY VIK MUIZNIEKS 1/78
2 REM *** PRINTS OUT CUMULATIVE DATA ON TEACHER TESTS
3 REM ***
4 WIDTH 132
5 CLEAR 10000
6 F=0: S=0
10 OPENR "CUM.1"
15 DIM M$(13),A(13),B(13),C(13),D(13)
16 FOR I=1 TO 13
17 READ M$(I): B(I)=500: C(I)=0: D(I)=0
18 NEXT I
19 GOSUB 220
20 ON ERROR GOTO 210
21 F=0
22 IF C=1 THEN F=1
30 N$="" DSKIN N$
32 IF C=3 AND INSTR(N$,M$)<>0 THEN LPRINT N$
33 IF C=3 AND INSTR(N$,M$)<>0 THEN F=1
40 IF C=1 THEN LPRINT N$
45 FOR I=1 TO 13
50 K=0: DSKIN K
51 IF C=3 AND F<>1 THEN 70
52 A(I)=A(I)+K: D(I)=D(I)+K*K
53 IF K<B(I) THEN B(I)=K
54 IF K>C(I) THEN C(I)=K
55 IF F=0 THEN 70
60 LPRINT M$(I);K
70 NEXT I
71 IF C=3 AND F<>1 THEN 20 ELSE S=S+1
72 IF C=2 THEN 20
75 LPRINT
76 LPRINT
80 GOTO 20
90 DATA "TEST RELIABILITY ="
100 DATA "STANDARD DEVIATION ="
110 DATA "STANDARD ERROR OF MEASUREMENT ="
120 DATA "% + ALTERNATIVES ="
130 DATA "% ? ALTERNATIVES ="
140 DATA "% 0 ALTERNATIVES ="
150 DATA "% INCORRECT ITEM DIFFICULTY ESTIMATES ="
155 DATA "AVERAGE ITEM DIFFICULTY ="
160 DATA "% EXCEPTIONALLY GOOD QUESTIONS ="
170 DATA "% GOOD QUESTIONS ="
180 DATA "% MARGINAL QUESTIONS ="
190 DATA "% NON-DISCRIMINATING QUESTIONS ="
200 DATA "% POSSIBLY AMBIGUOUS QUESTIONS ="
210 RESUME 215
215 CLOSR
217 GOTO 500
220 PRINT "TYPE THE NUMBER OF YOUR CHOICE:"
230 PRINT "  1. EVERYTHING"
240 PRINT "  2. AVERAGES ONLY"
250 PRINT "  3. ONE TEACHER ONLY"
260 PRINT "  4. STOP"
270 C=0: INPUT C
290 IF C<1 OR C>4 THEN 220
300 IF C<>4 THEN 330
310 CLOSR
320 GOTO 1000
330 IF C<>3 THEN RETURN
340 PRINT "TEACHER'S NAME";
350 M$="": INPUT M$
360 IF M$="" THEN 340 ELSE RETURN
500 IF S=0 THEN S=1
501 LPRINT "AVERAGE VALUES"; TAB(50); "LOW VALUE"; TAB(70); "HIGH VALUE";
          TAB(90); "STAN. DEV "
502 LPRINT
503 LPRINT
505 FOR I=1 TO 13
510 LPRINT M$(I); INT(A(I)/S*100)/100;
515 LPRINT TAB(52); B(I); TAB(72); C(I);
516 LPRINT TAB(92); INT(SQR(S*D(I)-A(I)*A(I))*100/S)/100
520 NEXT I
525 LPRINT
526 LPRINT
527 LPRINT S; "TESTS HAVE BEEN GRADED"
530 ERASE A, B, C, D, M$: RESTORE
540 GOTO 6
1000 END
```

APPENDIX B

SAMPLE OUTPUT FROM DUMP1

TEXT FILE

```
"  "
"  "
"  "
"KEY TO STUDENT ANSWER DISTRIBUTION SYMBOLS:"
"  "
"  "
A * INDICATES THE CORRECT ANSWER TO EACH QUESTION.
"  "
A + INDICATES THAT AN INCORRECT ANSWER WAS AN EXCEPTIONALLY GOOD ALTERNATIVE TO INCLUDE IN THE QUESTION.  AT LEAST TWO-THIRDS
OF THE STUDENTS WHO CHOSE THE ALTERNATIVE RANKED IN THE BOTTOM-HALF OF THE CLASS.
"     YOU HAVE#"
NUM
ALTERNATIVE(S) MARKED WITH A +.  THIS IS#
NUM
% OF ALL THE POSSIBLE INCORRECT ALTERNATIVES ON THE TEST.
"  "
A ? INDICATES THAT AN INCORRECT ANSWER (OR THE QUESTION) MAY BE AMBIGUOUS.  STUDENTS WHO RANKED IN THE TOP-HALF OF THE
CLASS CHOSE THE ALTERNATIVE MORE OFTEN THAN STUDENTS WHO RANKED IN THE BOTTOM-HALF OF THE CLASS.
"     YOU HAVE#"
NUM
ALTERNATIVE(S) MARKED WITH A ?  THIS IS#
NUM
% OF ALL THE POSSIBLE INCORRECT ALTERNATIVES ON THE TEST.
"  "
"AN 'OMIT' MEANS THAT A STUDENT DID NOT ANSWER THE QUESTION, OR A STUDENT MISMARKED THE ANSWER CARD SO THAT THE ANSWER"
WAS NOT ONE OF THE POSSIBLE CHOICES.
PAGE
"  "
OBSERVATIONS ABOUT YOUR TEST
"  "
"     IF NO ONE CHOOSES AN INCORRECT ALTERNATIVE, THEN THAT ALTERNATIVE IS NOT VERY EFFECTIVE.  YOU HAD#"
NUM
INCORRECT
ALTERNATIVE(S) WHICH NO ONE CHOSE.  THIS IS#
NUM
% OF ALL THE POSSIBLE INCORRECT ALTERNATIVES ON THE TEST.
"     #"
NUM
OF THESE WERE IN EASY QUESTIONS#
"    #"
NUM
OF THESE WERE IN MEDIUM QUESTIONS    #
NUM
OF THESE WERE IN HARD QUESTIONS
```

TEXT FILE CONTINUED

```
"YOU SHOULD EXPECT TO HAVE SOME INCORRECT ALTERNATIVES WHICH NO ONE TAKES IN EASY QUESTIONS.  HOWEVER, YOU PROBABLY DON'T"
WANT A LARGE NUMBER OF INCORRECT ALTERNATIVES WHICH NO ONE TAKES IN MEDIUM OR HARD QUESTIONS.
"  "
"     THE DIFFICULTY COLUMN CONTAINS YOUR ESTIMATE OF HOW HARD YOU THOUGHT EACH QUESTION WOULD BE.  IF YOU DID NOT ESTIMATE"
"THE DIFFICULTY OF A QUESTION, OR IF IT TURNS OUT THAT THE CALCULATED DIFFICULTY IS NOT THE SAME AS YOUR ESTIMATE, THEN"
THE CALCULATED DIFFICULTY IS PRINTED IN PARENTHESES.
"  "
"     AN EASY QUESTION IS ONE WHICH MORE THAN 75% OF THE STUDENTS GET CORRECT."
"     A MEDIUM QUESTION IS ONE WHICH 25-75% OF THE STUDENTS GET CORRECT."
"     A HARD QUESTION IS ONE WHICH LESS THAN 25% OF THE STUDENTS GET CORRECT."
"  "
YOU ESTIMATED THE DIFFICULTY CORRECTLY ON#
NUM
% OF THE QUESTIONS ON THE TEST.
"  "
"    #"
NUM
OF THE QUESTIONS ON YOUR TEST WERE EASY.  THIS IS#
NUM
% OF ALL THE QUESTIONS ON THE TEST.
"    #"
NUM
OF THE QUESTIONS ON YOUR TEST WERE MEDIUM.  THIS IS#
NUM
% OF ALL THE QUESTIONS ON THE TEST.
"    #"
NUM
OF THE QUESTIONS ON THE TEST WERE HARD.  THIS IS#
NUM
% OF ALL THE QUESTIONS ON THE TEST.
"  "
"     THE DISCRIMINATION OF EACH QUESTION IS BASED ON HOW WELL THE TOP-HALF OF THE STUDENTS DO ON THE QUESTION RELATIVE"
TO THE BOTTOM-HALF OF THE STUDENTS.  THE DISCRIMINATION ESTIMATE IS NOT AN ESTIMATE OF HOW WELL THE QUESTION
"MEASURES KNOWLEDGE, BUT RATHER OF HOW WELL THE QUESTION SEPARATES THE BETTER-SCORING STUDENTS FROM THE POORER-SCORING"
"STUDENTS.  THE ESTIMATE TAKES INTO ACCOUNT EACH QUESTION'S DIFFICULTY.  HOWEVER, THE ESTIMATE IS BASED ON"
"THE QUESTION AS A WHOLE, AND IT DOES NOT TAKE INTO ACCOUNT THE EXISTENCE OF ANY QUESTIONABLE INCORRECT ALTERNATIVES"
"(INDICATED BY ?'S).  THEREFORE, IT IS POSSIBLE TO GET AN 'EXCEPTIONALLY GOOD DISCRIMINATION' ESTIMATE AND STILL"
HAVE SOME INCORRECT ALTERNATIVES MARKED WITH ?'S.
"  "
""  'EXCEPTIONALLY GOOD DISCRIMINATION' MEANS THAT MOST OF THE STUDENTS WHO GOT THE QUESTION RIGHT WERE IN THE TOP-HALF OF THE CLASS
"        YOU HAD#"
NUM
QUESTION(S) MARKED AS HAVING EXCEPTIONALLY GOOD DISCRIMINATION.  THIS IS#
NUM
% OF ALL THE QUESTIONS ON THE TEST.
"  "
```

TEXT FILE CONTINUED

```
"    'GOOD DISCRIMINATION' MEANS THAT THE QUESTION DOES AN ADEQUATE JOB OF SEPARATING THE BETTER STUDENTS FROM THE POORER STUDENTS "
"        YOU HAD#"
NUM
QUESTION(S) MARKED AS HAVING 'GOOD DISCRIMINATION'.  THIS IS#
NUM
% OF ALL THE QUESTIONS ON THE TEST
" "
"    'MARGINAL DISCRIMINATION' MEANS THAT THE QUESTION JUST BARELY SEPARATES THE BETTER STUDENTS FROM THE POORER STUDENTS "
"        YOU HAD#"
NUM
QUESTION(S) MARKED AS HAVING 'MARGINAL DISCRIMINATION'.  THIS IS#
NUM
% OF ALL THE QUESTIONS ON THE TEST
" "
"    'DOES NOT DISCRIMINATE BETWEEN STUDENTS' MEANS THAT ALL STUDENTS DO EQUALLY WELL ON THAT QUESTION (REGARDLESS OF THEIR SCORE). "
"        YOU HAD#"
NUM
QUESTION(S) MARKED AS NOT DISCRIMINATING BETWEEN STUDENTS.  THIS IS#
NUM
% OF ALL THE QUESTIONS ON THE TEST
" "
""   'DISCRIMINATES IN THE WRONG DIRECTION' MEANS THAT THE POORER STUDENTS GOT THE QUESTION RIGHT MORE OFTEN THAN THE BETTER STUDENTS
"        YOU HAVE#"
NUM
QUESTION(S) MARKED AS DISCRIMINATING IN THE WRONG DIRECTION.  THIS IS#
NUM
% OF ALL THE QUESTIONS ON THE TEST
" "
"     IF A QUESTION DOES NOT DISCRIMINATE VERY WELL, IT MAY STILL BE A VERY GOOD QUESTION.  IT IS POSSIBLE THAT THE QUESTION"
"MEASURES SOMETHING DIFFERENT FROM WHAT THE MAJORITY OF THE TEST MEASURES.  HOWEVER, IT MIGHT BE A GOOD IDEA TO LOOK AT"
SUCH QUESTIONS TO SEE IF THERE MIGHT BE SOME PROBLEMS WITH THEM.  ANY QUESTIONS THAT DISCRIMINATE IN THE WRONG DIRECTION
SHOULD BE EXAMINED TO SEE IF THEY ARE AMBIGUOUS (OR IF THE ANSWER IS WRONG).
" "
*** WARNING ***  BLIND ACCEPTANCE (OR THOUGHTLESS DISREGARD) OF THE ABOVE RESULTS COULD BE HAZARDOUS TO YOUR TESTING PRACTICES!!
"     THE COMMENTS MADE ON THIS PRINTOUT ARE BASED ON (GENERALLY SOUND) TEST-ANALYSIS THEORIES.  THE PURPOSE OF TEST-ANALYSIS IS TO"
POINT-OUT TO YOU THE POSSIBLE STRENGTHS AND WEAKNESSES OF YOUR TEST.  ALTHOUGH IT IS POSSIBLE TO MAKE GENERAL STATEMENTS ABOUT
"YOUR TEST, YOU ARE THE ONLY ONE WHO KNOWS YOUR STUDENTS, YOUR CLASSROOM, AND THE PARTICULAR CIRCUMSTANCES UNDER WHICH THE TEST"
"WAS GIVEN.  THEREFORE, YOU MUST BE THE ONE TO DECIDE WHETHER OR NOT THE COMMENTS MADE ON THIS PRINTOUT ARE RELEVANT TO"
YOUR PARTICULAR TEST.
"     ALWAYS KEEP IN MIND THAT THE ABOVE RESULTS ARE NOT ABSOLUTE.  TESTS, AND TEST QUESTIONS, DO NOT HAVE UNALTERABLE"
"CHARACTERISTICS.  THE SAME TEST, GIVEN TO A DIFFERENT GROUP OF STUDENTS, OR EVEN TO THE SAME GROUP OF STUDENTS ON ANOTHER DAY,"
"COULD HAVE QUITE DIFFERENT RESULTS.  TEST-ANALYSIS IS NOT DESIGNED TO HELP YOU BUILD 'PERFECT' TESTS, BUT RATHER TO AID YOU IN"
"GAINING AN UNDERSTANDING OF THE TESTS YOU GIVE.  HOPEFULLY, YOU CAN USE THAT UNDERSTANDING TO IMPROVE YOUR FUTURE TESTS."
```